

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
<small>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</small> PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 07-05-2010		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) DEC 2010 - MAY 2010	
4. TITLE AND SUBTITLE Design, Build and Test of an Axial Flow Hydrokinetic Turbine with Fatigue Analysis.				5a. CONTRACT NUMBER N62271-97-G-0026	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Jerod W. Ketcham				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Massachusetts Institute of Technology				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943				10. SPONSOR/MONITOR'S ACRONYM(S) NPS	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT 1. DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT OpenProp is an open source propeller and turbine design and analysis code that has been in development since 2007 by MIT graduate students under the supervision of Professor Richard Kimball. In order to test the performance predictions of OpenProp for axial flow hydrokinetic turbines, a test fixture was designed and constructed, and a model scale turbine was tested. Tests were conducted in the MIT water tunnel for tip speed ratios ranging from 1.55 to 7.73. Additional code was also written and added to OpenProp in order to implement ABS steel vessels rules for propellers and calculate blade stress. The blade stress code was used to conduct a fatigue analysis for a model scale propeller using a quasi-steady approach. Turbine test results showed that OpenProp provides good performance predictions for the ondesign operational condition but that further work is needed to improve performance predictions for the off-design operational condition. Fatigue analysis results show that reasonable estimates of propeller blade fatigue life can be obtained using a relatively simple method.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 94	19a. NAME OF RESPONSIBLE PERSON Sean Tibbitts, Educational Technician
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code) (831) 656-2319 civins@nps.edu

Design, Build and Test of an Axial Flow Hydrokinetic Turbine with Fatigue Analysis

by
Jerod W. Ketcham

B.S., Mechanical Engineering
Wichita State University, 1997

Submitted to the Department of Mechanical Engineering and the Department of Materials Science and
Engineering in Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer

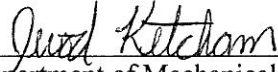
and

Master of Science in Materials Science and Engineering


at the Massachusetts Institute of Technology
June 2010

© 2010 Massachusetts Institute of Technology
All rights reserved

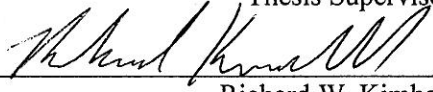
Signature of Author _____


Department of Mechanical Engineering
May 7, 2010


Certified by _____


Mark S. Welsh
Professor of the Practice of Naval Architecture and Engineering
Thesis Supervisor

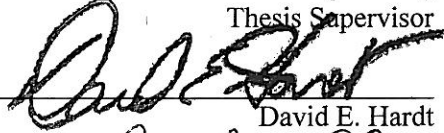
Certified by _____


Richard W. Kimball
Thesis Supervisor

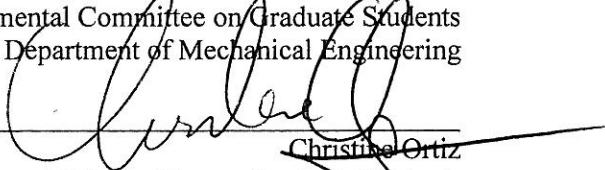
Certified by _____


Ronald G. Ballinger
Professor of Materials Science and Engineering and Nuclear Science and Engineering
Thesis Supervisor

Accepted by _____


David E. Hardt
Chair, Departmental Committee on Graduate Students
Department of Mechanical Engineering

Accepted by _____


Christine Ortiz
Chair, Departmental Committee on Graduate Students
Department of Materials Science and Engineering

Design, Build and Test of an Axial Flow Hydrokinetic Turbine with Fatigue Analysis

by

Jerod W. Ketcham

Submitted to the Department of Mechanical Engineering and the Department of Materials Science and Engineering on May 7, 2010 in partial fulfillment of the requirements for the degrees of Naval Engineer and Master of Science in Materials Science and Engineering

Abstract

OpenProp is an open source propeller and turbine design and analysis code that has been in development since 2007 by MIT graduate students under the supervision of Professor Richard Kimball. In order to test the performance predictions of OpenProp for axial flow hydrokinetic turbines, a test fixture was designed and constructed, and a model scale turbine was tested. Tests were conducted in the MIT water tunnel for tip speed ratios ranging from 1.55 to 7.73. Additional code was also written and added to OpenProp in order to implement ABS steel vessels rules for propellers and calculate blade stress. The blade stress code was used to conduct a fatigue analysis for a model scale propeller using a quasi-steady approach.

Turbine test results showed that OpenProp provides good performance predictions for the on-design operational condition but that further work is needed to improve performance predictions for the off-design operational condition. Fatigue analysis results show that reasonable estimates of propeller blade fatigue life can be obtained using a relatively simple method. Calculated blade stress distributions agree with previously published data obtained with more sophisticated and time consuming calculation techniques.

Thesis Supervisor: Mark S. Welsh

Title: Professor of the Practice of Naval Architecture and Engineering

Thesis Supervisor: Richard W. Kimball

Thesis Supervisor: Ronald G. Ballinger

Title: Professor of Materials Science and Engineering and Nuclear Science and Engineering

Acknowledgements

The author thanks the following individuals for their assistance in completing this thesis:

Professor Rich Kimball for his advice and guidance on this project. His enthusiasm for this project and practical advice were greatly appreciated.

Professor Chryssostomos Chryssostomidis for his interest and financial support for this project. Without funding this project would have been impossible. I am grateful for his patience and understanding when things did not always go according to plan.

Dr. Brenden Epps for his assistance with the MATLAB® code, work to restore the water tunnel and discussions where I could ask the simple questions. The duration of this project would have exceeded the time available without his help.

CAPT Mark Welsh for his leadership of the 2N program and encouragement during this project.

Professor Ron Ballinger for his practical insight and level of interest in this project. Fatigue analysis is a tricky subject and could not have been navigated without his assistance.

Most importantly, my family, Jill, Justin, Jessica, Jason and Janae, for all their support and encouragement. The long and sometimes frustrating days would not have been possible without their full support.

TABLE OF CONTENTS

Introduction	7
Chapter 1 –Development, Capability and Limitations of OpenProp	8
Development of OpenProp	8
Capability of OpenProp	8
Limitations of OpenProp	9
Chapter 2 – Hydrokinetic Turbine Design and Construction	10
Chapter 3 – Test Procedure, Results and Comparison	12
Test Procedure	12
Results and Comparison	16
Chapter 4 – Implementation of ABS Steel Vessel Rules for Blade Thickness	18
Rule Implementation in OpenProp	19
Limitations	19
Moment of Inertia Calculation	19
Chapter 5 – Calculation of Blade Stress	21
Theory	21
Implementation	25
Results	27
Limitations	30
Chapter 6 – Fatigue Analysis	31
Cyclic Load	31
Fatigue Failure	36
Limitations	39
Chapter 7 – Test Fixture Design and Construction	40
MECHANICAL	40
Thrust/Torque Sensor	40
Output Shaft Configuration	42
Drive Shaft Configuration	43
Housings	44
ELECTRICAL	45
Slip Rings	45
Amplifiers	46
Motor	46
Controller	48
CONSTRUCTION	51
Chapter 8 – Conclusions and Recommendations	53
Conclusions	53
Recommendations for Further Work	53
References	54
Appendix A – Codes	55
Appendix B – Parts List	76
Appendix C – Drawings	78

TABLE OF FIGURES

Figure 1: Turbine Drawing <i>Courtesy of Epps</i>	11
Figure 2: Turbine Test	11
Figure 3: Thrust Calibration.....	12
Figure 4: Torque Calibration	13
Figure 5: Spectrum Analysis–600RPM, 1.69m/s	13
Figure 6: Friction Torque.....	14
Figure 7: Measured Torque – No Correction.....	14
Figure 8: Test Section Flow Speed Determination	16
Figure 9: Results	16
Figure 10: Blade Section with Lift and Flow Velocity Vectors	21
Figure 11: Blade Section Showing Lift Resolved into Axial and Tangential Components	22
Figure 12: Bending Moments Components	23
Figure 13: Total Bending Moments about Centroidal Axes.....	24
Figure 14: Example Propeller <i>Courtesy of Epps</i>	25
Figure 15: Distorted Root Section	25
Figure 16: Undistorted Root Section	26
Figure 17: Calculation of Elemental Area Properties	26
Figure 18: On-design Root Section Stress.....	27
Figure 19: On-design Suction Side Stress: $J_s=0.75$, $V_s=1.5\text{m/s}$, $n=8\text{rev/s}$, $D=0.25\text{m}$	28
Figure 20: On-design Pressure Side Stress: $J_s=0.75$, $V_s=1.5\text{m/s}$, $n=8\text{rev/s}$, $D=0.25\text{m}$	28
Figure 21: Off-design Suction Side Stress: $J_s=0.40$, $V_s=1.5\text{m/s}$, $n=15\text{rev/s}$, $D=0.25\text{m}$	29
Figure 22: Off-design Pressure Side Stress: $J_s=0.40$, $V_s=1.5\text{m/s}$, $n=15\text{rev/s}$, $D=0.25\text{m}$	29
Figure 23: Sector, Single Screw Ship Wake	32
Figure 24: Original Inflow Velocities.....	32
Figure 25: New Inflow Velocities.....	32
Figure 26: Pressure Side Blade Stress for Each Wake Sector	33
Figure 27: Point of Maximum Tensile Stress	34
Figure 28: Maximum Blade Stress versus Angular Position for Various Ship Speeds Using the On-Design Advance Coefficient ($J_s=0.75$).....	35
Figure 29: S-N Curve for NiAl Bronze	36
Figure 30: Operational Profile for DDG51	36
Figure 31: Example Operational Profile Used for Calculations	37
Figure 32: Time at Various Stress Levels.....	37
Figure 33: Provided Sensor.....	41
Figure 34: Stress from Axial Load on Sensor (50lbf applied).....	41
Figure 35: Stress from Torque Load on Sensor (12ft-lbf applied)	42
Figure 36: Output Shaft Configuration	43
Figure 37: Driveshaft and Bearing Assembly with Brush Blocks and Slip Rings	44
Figure 38: Installed Slip Rings and Brushes.....	45
Figure 39: K089300-7Y Torque Speed Curve.....	47
Figure 40: Output Power Capability	48
Figure 41: Limiting Current.....	48
Figure 42: Electrical Components	50
Figure 43: Schematic of Enclosure Electrical Components	51
Figure 44: Completed Test Fixture in Operation.....	52

TABLE OF TABLES

Table 1: Development History of OpenProp	8
Table 2: Key Turbine Parameters	11
Table 3: Test Tip Speed Ratios.....	15
Table 4: Test Fixture Limitations	40

Introduction

Since 2007, graduate students at the Massachusetts Institute of Technology (MIT) have been developing an open source propeller and turbine design and analysis tool under the supervision of Professor Richard Kimball. The tool is a set of open source MATLAB® scripts published under the GNU General Public License which are capable of performing design and analysis studies for open and ducted propellers as well as axial flow turbines. This suite of MATLAB® scripts is called OpenProp. OpenProp propeller design capabilities include performing parametric studies of propellers using various propeller diameters, number of blades and rotation speeds. Propeller analysis features include performing off-design and cavitation analyses. A gap in OpenProp capabilities was the inability to evaluate the structural adequacy of a propeller or turbine. This project added two new modules. One module implements American Bureau of Shipping (ABS) steel vessel rules for propellers and the other calculates the blade surface stress.

Validation of OpenProp turbine and propeller performance predictions is limited. The portion of the code suite which designs ducted propellers has been validated against the US Navy's Propeller Lifting Line (PLL) code with excellent correlation. Several experiments have been done to validate open propeller performance predictions using a modified trolling motor apparatus. One test had been performed, with limited success, of an axial flow turbine. No tests had been performed for ducted propellers. Because of this lack of experimental validation of OpenProp, it became necessary to design and construct a propeller and turbine test fixture that is robust and can easily be used to test open and ducted propellers as well as turbines. This project provided a test fixture, funded by MIT SeaGrant, which can be used in a water tunnel or tow tank to provide experimental performance results which can be used to validate OpenProp performance predictions.

OpenProp implements the vortex lifting line method to quickly achieve a propeller or axial flow turbine design. The lifting line method of propeller design has some limitations but is an excellent method to obtain an initial design which can be refined using more sophisticated design techniques. In the spirit of providing initial design estimates, this project also completed a quasi-steady fatigue analysis and predicted the fatigue life of a propeller.

This paper presents the results of testing, blade stress calculations and fatigue analysis.

Chapter 1 –Development, Capability and Limitations of OpenProp

Development of OpenProp

OpenProp had its genesis in a code called MATLAB® Propeller Vortex Lattice (MPVL) which was a code which added graphical user interfaces to Propeller Vortex Lattice (PVL) which was developed by Kerwin (2007) for his propellers course at the Massachusetts Institute of Technology (MIT). Since that time significant capability has been added to the code and additional features and capability are being developed.

OpenProp uses a lifting line method to model blade circulation (Kerwin 2007). The lifting line technique has been well established and was implemented by Kerwin for preliminary parametric propeller design for the US Navy in a code called PLL. OpenProp development sought to expand and enhance the capabilities of Kerwin’s code and make the software more user friendly. A full explanation for the theory of operation of OpenProp has been given by Epps (2010b).

Capability of OpenProp

A table showing the development history and current capability of OpenProp is shown below.

Date	Event	Persons Responsible	Description
2001	PVL Developed	J.E. Kerwin	Lifting line design code used for Kerwin’s propeller class at MIT
2007	MPVL Developed (Later renamed OpenProp v1.0)	H. Chung K.P. D’Epagnier	MATLAB® version of PVL which incorporated GUIs for parametric and blade row design and geometry routines for CAD (Rhino) interface. This code used a Lerb’s criteria optimizer. Chung (2007), D’Epagnier (2007)
2008	Cavitation Analysis Routines Developed	C.J. Peterson	Using Drela’s XFOIL, routines and executables were developed for conducting propeller cavitation analysis, Peterson (2008).
2008	OpenProp v2.0	J.M. Stubblefield	Added capability for ducted propeller design Stubblefield (2008)
2009	OpenProp v2.3	B.P. Epps	Added new optimizer. Incorporated routines of Peterson, added off-design analysis, corrected errors and added ability to design axial flow turbines with or without blade chord optimization. Theory described in Epps (2010b).
2010	Contra-Rotating Propeller Design	D. Laskos	Added the capability for contra-rotating propeller design with cavitation analysis. Laskos (2010)

Table 1: Development History of OpenProp

This project added the capability to calculate blade stress and implement ABS rules for propellers. Epps continues to refine and expand OpenProp capabilities and is currently working on codes to predict propeller performance during shaft reversals.

Limitations of OpenProp

OpenProp uses the lifting line method to model the blade circulation. There are limits in regard to using this method in propeller design.

1. Constant Radius Vortex Helix – In the implementation of the lifting line method, the trailing vorticity is assumed to be of constant radius. For propellers, it is known that the trailing vorticity helix radius actually decreases. This simplification has been made to ease the complexity of calculating the influence of the trailing vorticity on the blade itself. The errors introduced with this simplification are relatively minor as shown in the experimental data comparison in this paper and by Stubblefield (2008) in his comparison of OpenProp predictions to a more established propeller design code.
2. Skew and Rake – OpenProp does not allow the designer to design blades with skew or rake
3. 3D Lifting Surface Effects – By its very nature a design tool based on the lifting line method cannot account for the 3D lifting surface effects. The result is that the translation from calculated hydrodynamic characteristics to a blade geometry which produces the desired hydrodynamic performance is difficult and in most cases will contain errors. Therefore, the blade geometry that is generated from a lifting line code is an approximation and one should not expect the propeller performance measured by experiment to completely match the lifting line predicted performance. A comparison and discussion of predicted performance versus experimental performance for *U.S. Navy Propeller 4119* is contained in Epps (2010a, p. 224-225).

Chapter 2 – Hydrokinetic Turbine Design and Construction

In propeller design the overall objective is to generate a specified thrust while minimizing the torque required to produce it. In turbine design the goal is to maximize the torque and minimize the thrust. A procedure which can be used with OpenProp for turbine design is:

1. Determine expected C_D and C_L . Typical ranges for these quantities are $0.008 < C_D < 0.03$ and $0.2 < C_L < 0.5$. The actual values for these parameters are dictated by the choice of blade section shape, flow regime and the degree of blade section scaling.
2. Perform parametric design study using expected C_D/C_L to determine number of blades and tip speed ratio. A typical value for this ratio is 0.06.
3. Select a design point from the parametric study of step 2. The turbine design point is characterized by the number of turbine blades and the tip speed ratio. In general, the more blades that a turbine has the greater its efficiency. However, in actual application this must be balanced by the manufacture costs of the turbine.
4. Choose the turbine diameter, free stream flow speed and rotation rate consistent with the chosen tip speed ratio in step 3 above such that desired power is achieved. Maximum turbine diameter is dictated by the water depth and installation scheme where the turbine will operate. It is generally desirable to maximize the turbine diameter in order to maximize the turbine's power capacity. Free stream flow speed is determined by the flow where the turbine will be installed. Desired rotation rate will be effected by the electrical generator selected for use with the turbine.
5. Perform an off-design performance analysis. An off-design performance analysis is necessary to obtain an overall picture of the time average power that the turbine will produce. This analysis is especially important for tidal turbines where there is a fluctuation of flow speed.
6. Determine the span-wise blade chord and thickness distribution. This step is where the blade geometry is determined to produce the characteristics determined in the previous steps. OpenProp can perform this step automatically by using the chord optimizer.
7. Perform blade stress analysis. A blade stress analysis is necessary to ensure the structural adequacy of the blades.

The above procedure was used to design the turbine which was tested by Epps. Results are shown in Epps (2010a). The same turbine was retested as part of this project. Step 7 was not performed as part of the design process because the stress module of OpenProp was not available at that time. The turbine diameter was selected as the maximum diameter which could be manufactured using the available rapid prototyping equipment and tested in the water tunnel test section. The number of blades was also dictated by the desire to maximize the turbine diameter and two blades were selected.

Epps (2010a) describes the procedure implemented in OpenProp to conduct a parametric design study, optimize a single turbine design and perform off-design analysis. For turbines the method can be summarized as setting the blade circulation less than zero and then simultaneously solving a set of equations such that the resulting variables represent a physically realizable condition. Epps (2009, 2010a) also discusses the correct way to optimize a turbine design.

Once the turbine was designed, the geometry module of OpenProp was used to create the set of points that represent the blade surface in three dimensions. This set of points was loaded into SolidWorks® via a macro developed for this purpose. In SolidWorks®, the blade geometry was turned into a solid which was multiplied into two blades and attached to a hub. This file was saved in .stl format and loaded into the rapid prototyping machine for production. The model scale turbine that was generated in this way was tested by Epps and as part of this project. Turbine test results are presented in the next section.

Table 9.1 of Epps (2010a, p. 280) contains the turbine design parameters. That table is reproduced here in Table 2.

Parameter	Value	Description
Z	2	Number of blades
n	19.1 rev/s	Rotation rate
D	0.25 m	Diameter
V_s	3 m/s	Free stream speed
D_{hub}	0.08382	Hub diameter
M	20	Number of panels
ρ	1000 kg/m ³	Water density
λ	5	Tip speed ratio
$C_{L,max}$	0.5	Maximum allowable lift coefficient

Table 2: Key Turbine Parameters

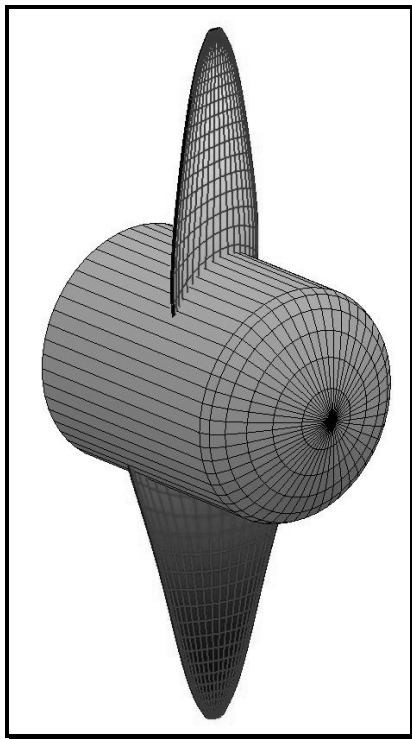


Figure 1: Turbine Drawing
Courtesy of Epps

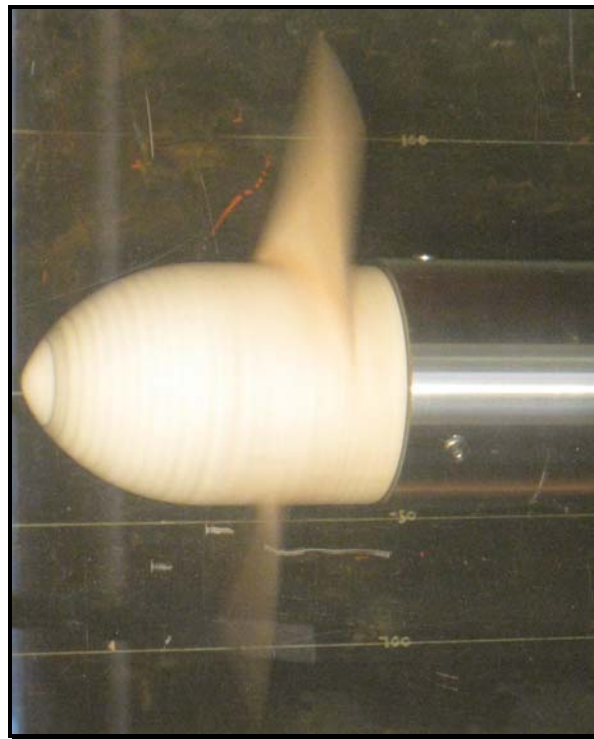


Figure 2: Turbine Test

Chapter 3 – Test Procedure, Results and Comparison

Test Procedure

Turbine testing was performed on a test fixture specifically designed for this purpose. The test fixture was funded via MIT SeaGrant and its design and construction are described in Chapter 7. Generally the test procedure consisted of measuring the shaft torque created by the turbine for various rotation rates and flow speeds. A detailed test procedure follows with test points in Table 3 and test results in Figure 9.

Calibration

Calibration of the test fixture was performed by hanging known weights from the output shaft of the test fixture and reading strain gage amplifier output voltage using LabView®. This calibration technique is a static calibration; a better calibration technique for this type of testing would have been a dynamic calibration. However, a dynamic calibration is more complicated and requires additional equipment which was unavailable. LabView® was connected to the test fixture in an identical way for both calibration and testing. Results of the calibration are shown Figure 3 and Figure 4.

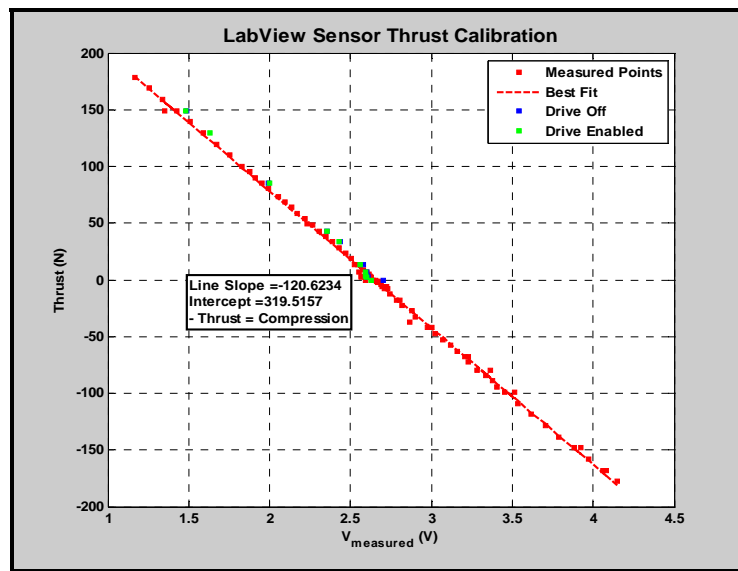


Figure 3: Thrust Calibration

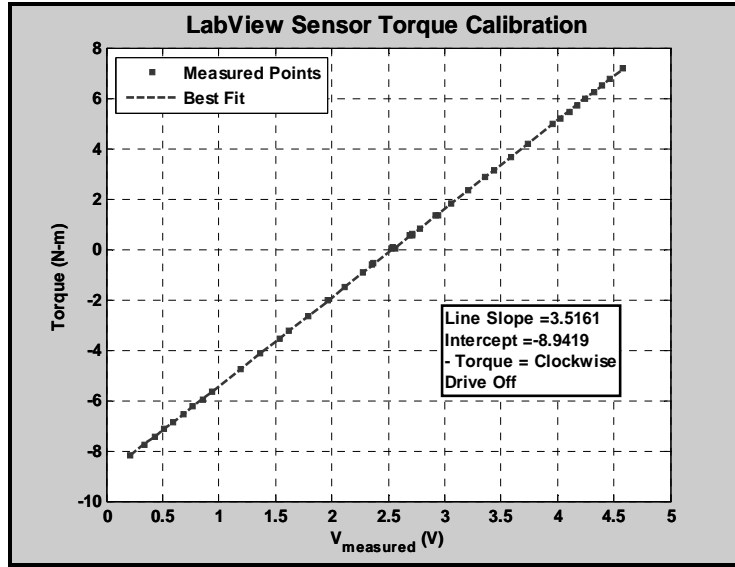


Figure 4: Torque Calibration

Because the motor drive used for these tests uses pulse width modulation (PWM) at 300VDC and because the signal wires are running alongside the power cable (inside the same 1.5 inch diameter standpipe) there was a concern that the Signal to Noise Ratio (SNR) would be too small to be able to effectively measure the signal voltage. This concern was allayed by performing spectral analyses on the measured signal. A typical result of these analyses is shown in Figure 5. The graph shows that there is minimal interference.

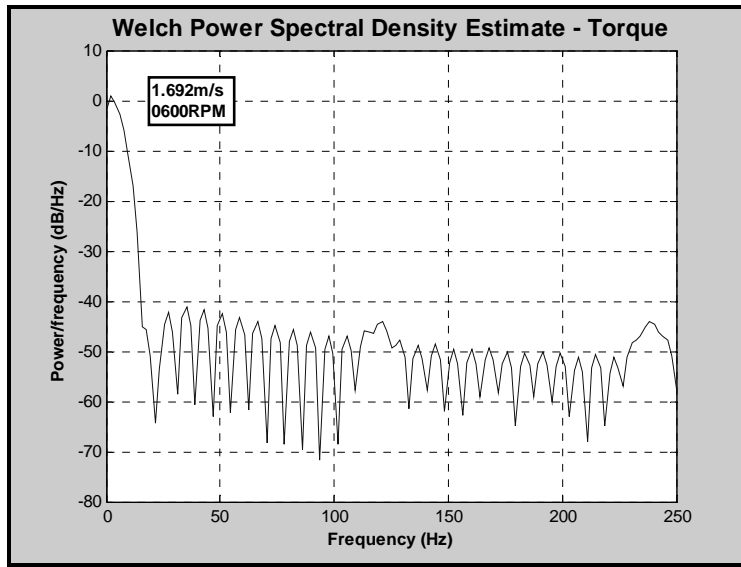


Figure 5: Spectrum Analysis-600RPM, 1.69m/s

Since the calibration that was used was a static calibration, it is necessary to correct the measured torque with the friction torque in order to determine the actual torque produced by the turbine. A graph of friction torque measured at various rotation rates without a hub or turbine attached, but

with the test fixture submerged in the test section, is given in Figure 6. These values are used to correct the torque measured by the sensor.

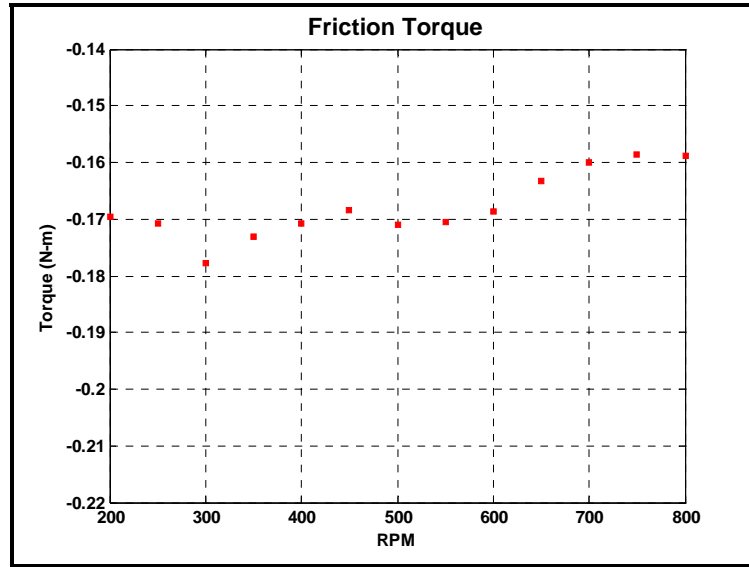


Figure 6: Friction Torque

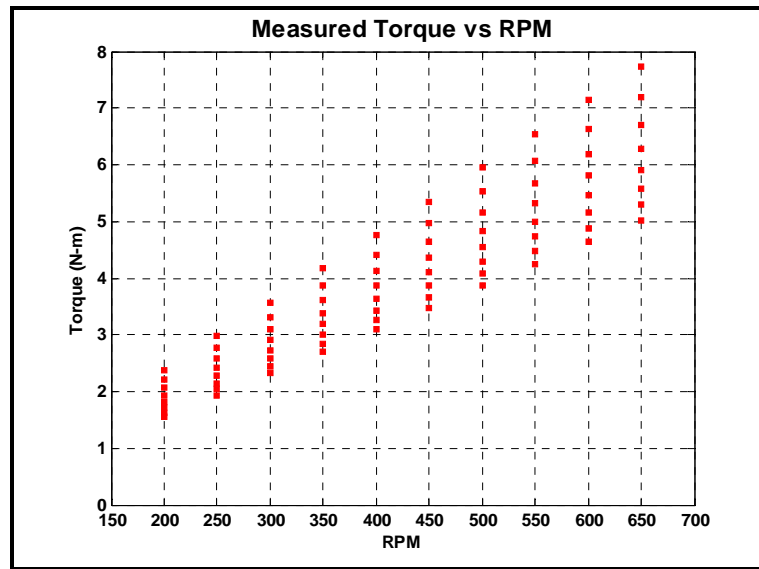


Figure 7: Measured Torque – No Correction

Figure 7 shows the torque measured by the sensor without torque correction. The data points of this figure represent the uncorrected torque values which were measured at the tip speed ratios listed in Table 3. Comparison of Figure 6 and Figure 7 shows that the friction torque is a relatively small value compared to the total measured torque.

Test Steps

Testing began by determining the tip speed ratios that would bracket the turbine's design point and reproduce the entire off-design performance curve generated by OpenProp. The tip speed ratios which were used in this test are shown in Table 3.

Tip Speed Ratio - λ										
Flow Speed (m/s)	RPM									
	200	250	300	350	400	450	500	550	600	650
	1.100	2.38	2.97	3.57	4.16	4.76	5.35	5.95	6.54	7.14
	1.185	2.21	2.76	3.31	3.87	4.42	4.97	5.52	6.08	6.63
	1.269	2.06	2.58	3.09	3.61	4.13	4.64	5.16	5.67	6.19
	1.354	1.93	2.42	2.90	3.38	3.87	4.35	4.83	5.32	5.80
	1.438	1.82	2.28	2.73	3.19	3.64	4.10	4.55	5.01	5.46
	1.523	1.72	2.15	2.58	3.01	3.44	3.87	4.30	4.73	5.16
	1.608	1.63	2.04	2.44	2.85	3.26	3.66	4.07	4.48	4.89
	1.692	1.55	1.93	2.32	2.71	3.09	3.48	3.87	4.25	4.64

Table 3: Test Tip Speed Ratios

The steps taken to gather the data displayed in Figure 9 are outlined below:

1. Generate Table 3 which represents the test points at which data was gathered. Flow speeds selected correspond to integer speed reference number increments of Figure 8
2. Set water tunnel impeller speed to create desired flow speed in test section
3. Command desired test fixture motor rotation
4. Collect torque voltage measurements via the LabView® interface. Sample rate was set at 500Hz. Sample time was 5-10 seconds.
5. Increase test fixture motor rotation rate
6. Wait approximately 10 seconds for transient behavior to subside
7. Repeat steps 4-6 until data has been collected for every rotation rate at the test section flow speed
8. Increase test section flow speed
9. Wait approximately one minute for transient behavior to subside.
10. Repeat steps 3-9 until all data has been collected.

Conducting the test in the order listed above minimizes the time required to collect data since the transient is much longer for a water tunnel impeller speed change than for a test fixture motor speed change.

In step 2, the water flow speed in the tunnel was not measured directly. Normal mode of operation is to measure the flow speed in the test section using a Laser Doppler Velocimetry (LDV) system; however the LDV system was not operational at the time of the test. Previous experimentation in the water tunnel generated Figure 8. Figure 8 relates impeller rotation rate to test section flow speed. This data was gathered using a Particle Image Velocimetry (PIV) flow measurement technique with a trolling motor test apparatus in the test section. The trolling motor provides similar test section blockage as the test fixture described herein. Note that the speed reference number in Figure 8 corresponds to the output frequency from the impeller motor drive to the impeller motor.

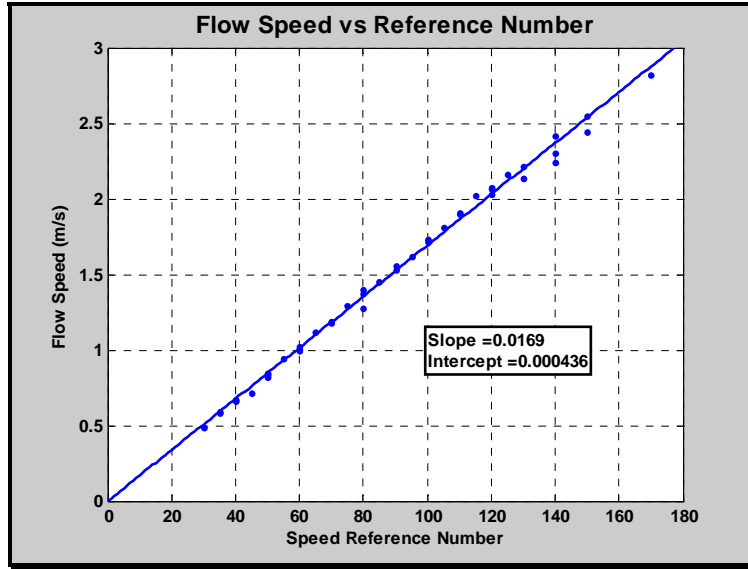


Figure 8: Test Section Flow Speed Determination

Step 3 was accomplished by operating the test fixture motor drive in the programmed velocity mode via the ASCII command line of the Copley Motion Explorer (CME) software. In the programmed velocity mode, a rotation speed is commanded and the motor drive maintains this speed regardless of the direction of energy flow. For this test the motor is acting as a generator being held at the commanded rotation rate. In the command window of CME it was observed that the RPM was being held to the commanded RPM \pm 2-3 RPM.

Results and Comparison

The results of the test are shown in Figure 9.

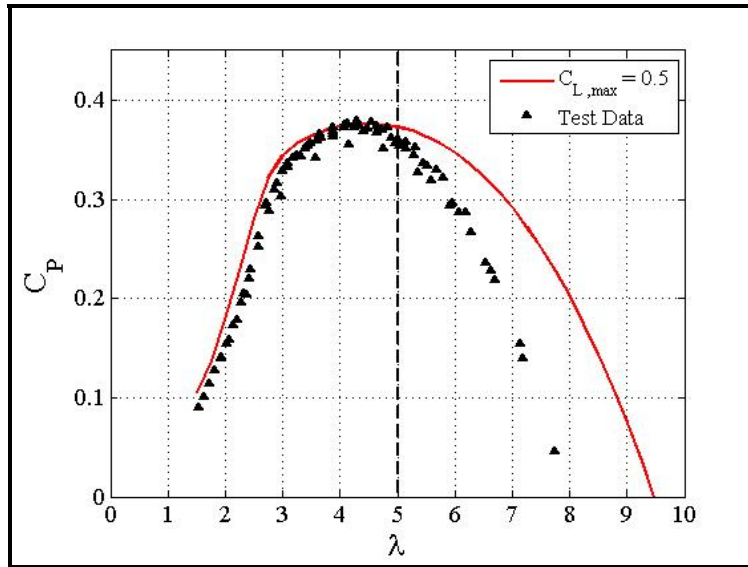


Figure 9: Results

Figure 9 shows the following:

1. There is good agreement between predicted and experimental data for tip speed ratios (λ) less than 5.
2. On-design predicted performance almost exactly matches the experimental on-design performance.
3. Experimental and predicted performance diverge for λ greater than 5.

As a result of the experimental results shown in Figure 9, OpenProp is being revised to more accurately predict performance for λ greater than λ_{Design} . It is thought that the divergence can be accounted for by implementing a more sophisticated model of bound and free circulation via lifting surface methods. This is a point of ongoing work in OpenProp.

Chapter 4 – Implementation of ABS Steel Vessel Rules for Blade Thickness

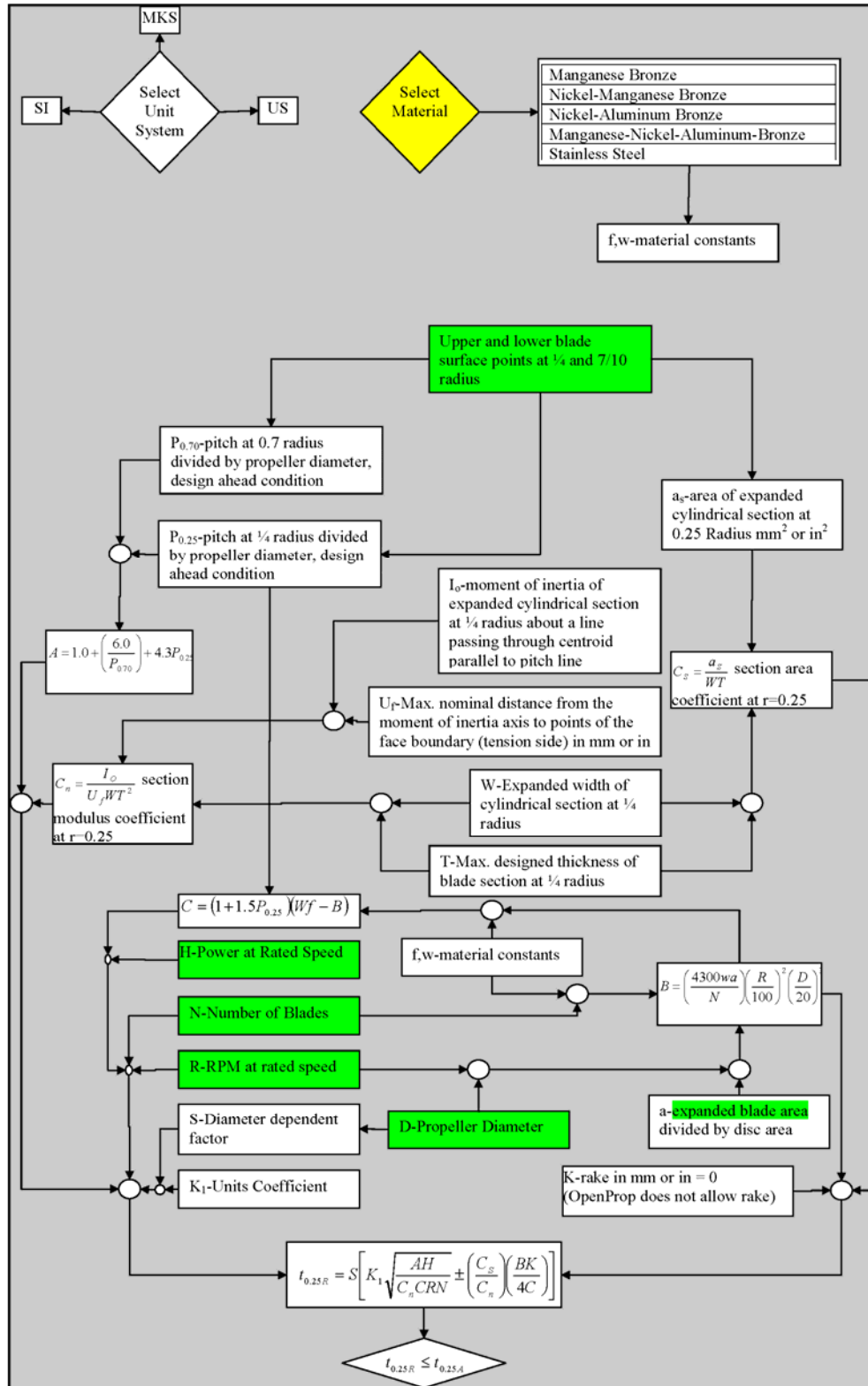


Figure 1: Variable Interrelationships from ABS Steel Vessel Rules for Propellers

This section describes the implementation of the American Bureau of Shipping (ABS) steel vessel rules into OpenProp as a first attempt in the design process to check the adequacy of the blade dimensions and material to support the loads they will carry. The output of OpenProp blade structure code is a check of the blade thickness at the quarter span section against the required blade thickness at the quarter span section as determined from implementation of the steel vessel rules. While the steel vessel rules do not actually calculate a stress or determine the operational lifetime of a propeller they do take these quantities into consideration as evidenced by the rules requirement to qualify a material other than those listed for service in a classed vessel. The ABS rules also represent what is generally required in order to class a vessel with any one of the many classification societies worldwide.

Rule Implementation in OpenProp

The OpenProp module which implements the ABS rules for propellers does so in a way which follows the flowchart shown in the figure above. User input for this module is only the material that is being used for the propeller construction. ABS lists five different materials that can be used for propeller manufacture; these are listed in the flowchart above. The lines of code which correspond to the desired material must be uncommented in order to use that material in the calculations. All other required input for implementation of the rules is automatically extracted from other modules of OpenProp or calculated within the blade structure module. User input is highlighted in yellow; input from other modules is highlighted in green. Since other OpenProp modules use the SI unit system, the user is not permitted to select a different unit system. The output of the structure module is a small table which lists the section thickness at the quarter span section and the required section thickness at the quarter span section, as calculated from the ABS rules. Propeller redesign is necessary if the required blade thickness is greater than the design blade thickness.

Limitations

In its current version, OpenProp designs fixed pitch, single propellers and turbines without rake or skew. The ABS rules for propellers allow for controllable pitch, rake and skew but the structure module developed as part of this project only performs the calculations for fixed pitch, single propellers without rake or skew. The rules used to develop the code of this project do not cover contra-rotating propellers, ducted propellers or propellers for vessels in ice. Additional structure capability could easily be added at a later date to incorporate the ever increasing capabilities of the OpenProp.

Moment of Inertia Calculation

The bulk of the code to implement the ABS rules for propellers is used to determine the moment of inertia of the designed propeller quarter span blade section. The blade structure module of OpenProp imports the points from the pressure and suction sides of the quarter span section. All of the points are then shifted so that the points lie in the first quadrant of the x-y plane. Shifting the points makes the determination of the quarter span section neutral axis easier. The code then performs a trapezoidal integration for the pressure and suction sides separately and subtracts the area of the pressure side from the suction side so that only the enclosed section area remains. The moment of inertia about the x-axis is then calculated and the parallel axis theorem used to

find the moment of inertia about the neutral axis. A flowchart of the portion of the code which calculates the moment of inertia is shown in the figure below.

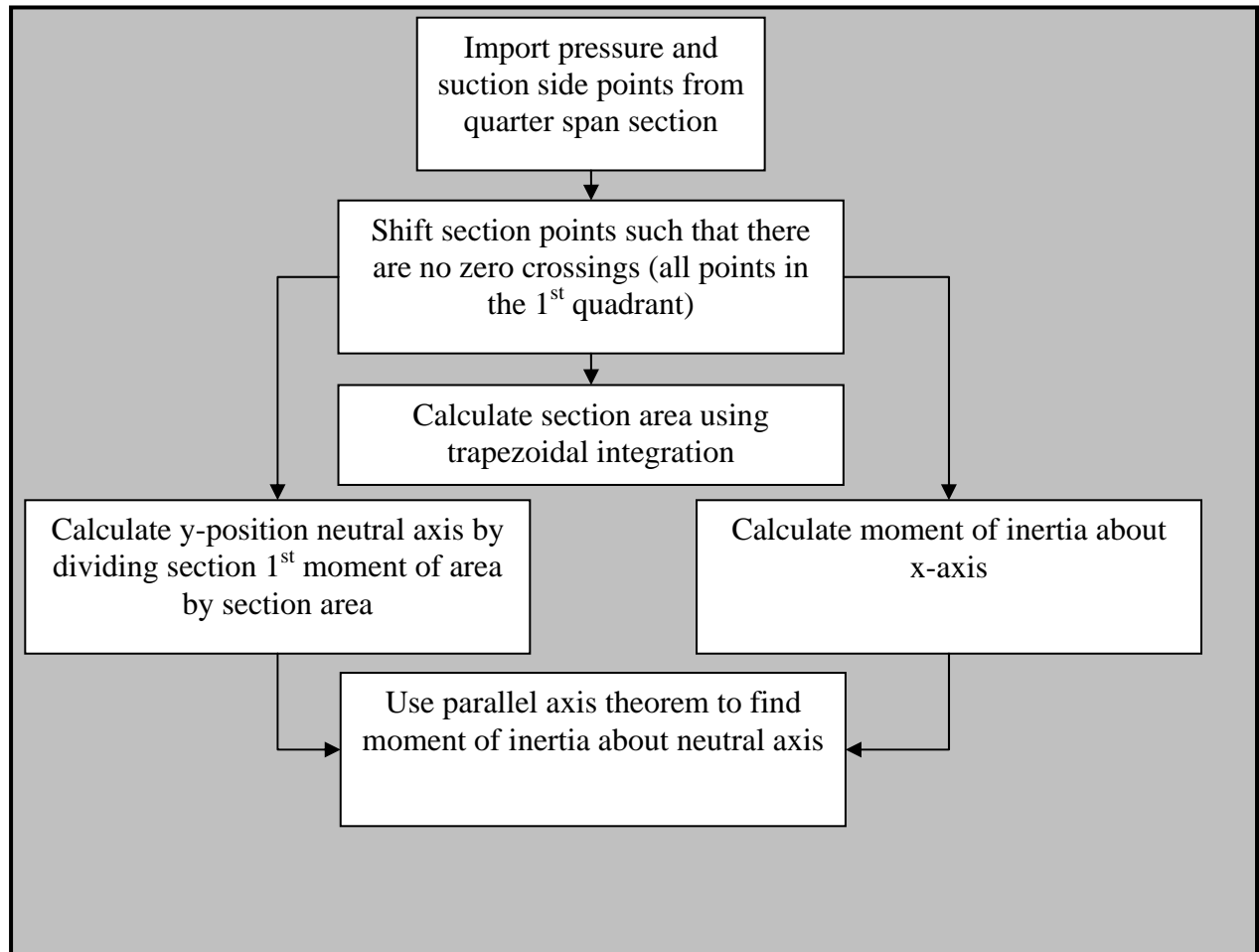


Figure 2: Code Flowchart to Find Section Area and Moment of Inertia

Chapter 5 – Calculation of Blade Stress

Theory

A relatively simple method to estimate the stress on a propeller blade is to implement beam bending theory. The derivation given below is an amplification of the derivation presented in Kerwin and Hadler (2010). Kerwin and Hadler also include some historical background for this method. The basic assumptions of the derivation are:

1. The blade acts as a cantilevered beam.
2. Axial stresses are due to bending and centrifugal forces.
3. Shear stresses are negligible.

Figure 10 below shows a propeller blade section with the associated inflow velocities and lift force. By definition the lift force, dL , is always perpendicular to the total inflow velocity V^* . dL is responsible for both thrust and torque on the propeller blades and propeller shaft.

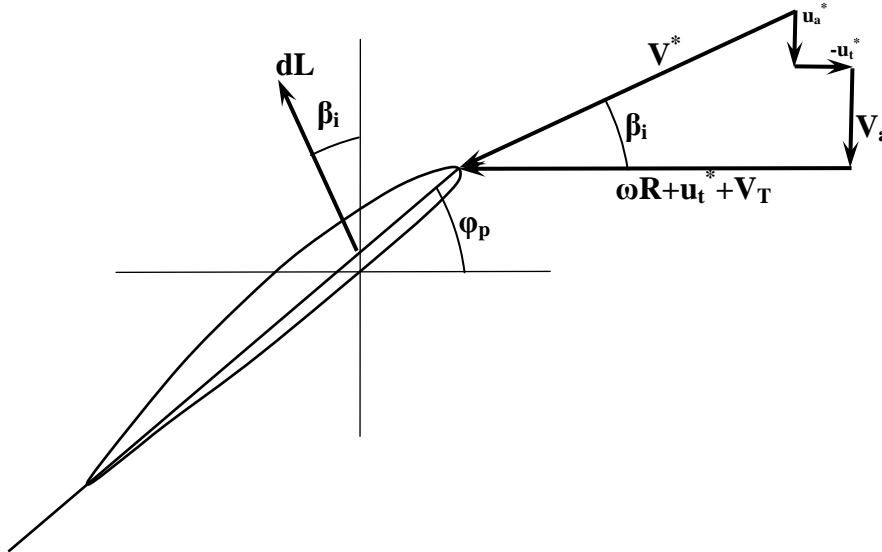


Figure 10: Blade Section with Lift and Flow Velocity Vectors

Note that dL is always perpendicular to V^* but it is typically not perpendicular to the chord line. Therefore, when determining the component of dL that produces thrust and the component of the dL which produces torque, the inflow angle β_i is required, not the blade pitch angle, ϕ_p . The elemental lift at a blade section is given by Equation 1.

$$dL = \frac{1}{2} \rho (V^*)^2 C_L c dr \quad (1)$$

where

dL = elemental lift on a blade section

ρ = fluid density

C_L = section lift coefficient at radius r , this comes from the lifting line calculation in OpenProp.

c = section chord length at r

dr = elemental radial span

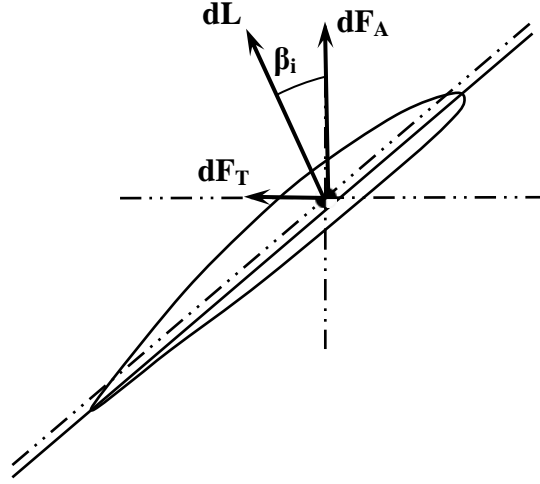


Figure 11: Blade Section Showing Lift Resolved into Axial and Tangential Components

From Figure 11, it can be seen that the axial force, F_A , and tangential force, F_T , at a blade section are given by:

$$dF_A = dL \cos(\beta_i + \varepsilon) \quad (2)$$

$$dF_T = dL \sin(\beta_i + \varepsilon) \quad (3)$$

where

β_i = inflow angle

$\varepsilon = C_D/C_L$, inflow angle correction due to viscous effects

C_D = section drag coefficient

Note that in Figure 11, the point of application of dL has been shifted to the centroid of the section and is no longer located at the same point as in Figure 10. This is done to simplify calculations. dL will not necessarily be located at the section centroid but at a point approximately $\frac{1}{4}$ of the distance from the leading edge to trailing edge on the chord line as shown in Figure 10. The fact that dL does not act through the section centroid means that dL will produce a torque about the span line of the blade. This torque and its associated shear stress are assumed to be negligible along with all other shear stresses.

Both F_A and F_T produce bending moments about the centroidal axes. Each of these bending moments, along with their x and y components, is shown in Figure 12. The equations for the bending moment are:

$$dM_A = (r - r_o) dF_A \quad (4)$$

$$dM_T = (r - r_o) dF_T \quad (5)$$

where

r_o = section radius where dM is being calculated

r = radius of section producing lift

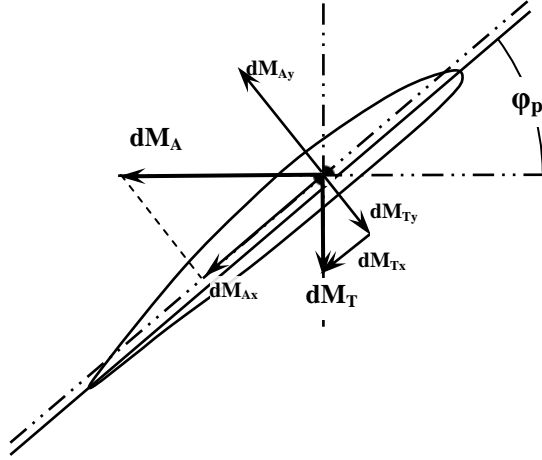


Figure 12: Bending Moments Components

The total moments produced by F_A and F_T , at a section r_o , are given by:

$$M_A = \int_{r_o}^R \frac{1}{2} \rho (V^*)^2 C_L c \cos(\beta_i + \varepsilon) (r - r_o) dr \quad (6)$$

$$M_T = \int_{r_o}^R \frac{1}{2} \rho (V^*)^2 C_L c \sin(\beta_i + \varepsilon) (r - r_o) dr \quad (7)$$

Because it is necessary to project these bending moments onto the centroidal axes of the section, blade pitch angle is required. Projecting the total bending moments onto the centroidal axes, the equations become

$$M_{X_o} = M_A \cos(\varphi_p) + M_T \sin(\varphi_p) \quad (8)$$

$$M_{Y_o} = M_A \sin(\varphi_p) - M_T \cos(\varphi_p) \quad (9)$$

Each of these bending moment vectors is shown in Figure 13.

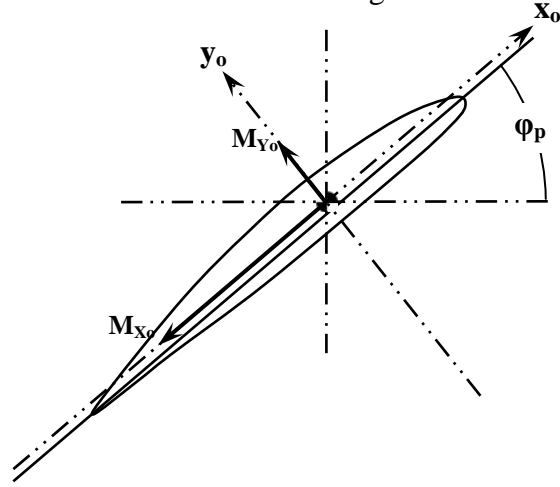


Figure 13: Total Bending Moments about Centroidal Axes

Additionally, the centrifugal force acting at each section contributes to the overall stress at the section. The elemental centrifugal force acting on a blade from an adjoining section is given by

$$dF_C = (2\pi n)^2 r dm \quad (10)$$

where

$dm = \rho_b A dr$ = mass of blade element

ρ_b = propeller blade material density

A = section area

c = section chord length

t = section thickness

Summing the contributions of all adjoining sections to the F_C at the section of interest, the total F_C at the section becomes

$$F_C = (2\pi n)^2 \rho_b \int_{r_o}^R A r dr \quad (11)$$

Since the blades analyzed using the above method do not contain rake or skew, which would introduce additional bending moments from F_C , the equation for the stress on a blade section can be expressed as:

$$\sigma = \frac{-M_{x_o} y}{I_{x_o}} - \frac{M_{y_o} x}{I_{y_o}} + \frac{F_C}{A} \quad (12)$$

Implementation

The following paragraphs are an explanation of the method used to calculate blade stresses. The calculations were performed on a propeller which was designed by Epps (2010a) and is shown in Figure 14.

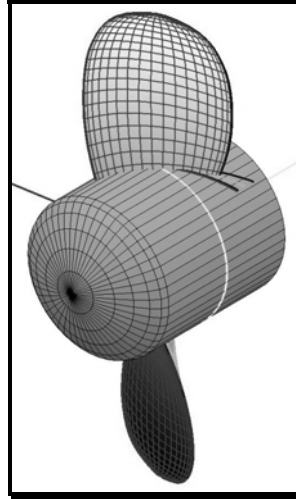


Figure 14: Example Propeller *Courtesy of Epps*

In order to implement the equations above it is necessary to calculate the required blade section quantities. Figure 15 and Figure 17 illustrate the overall procedure for determining 2D blade section area, centroid and moments of inertia.

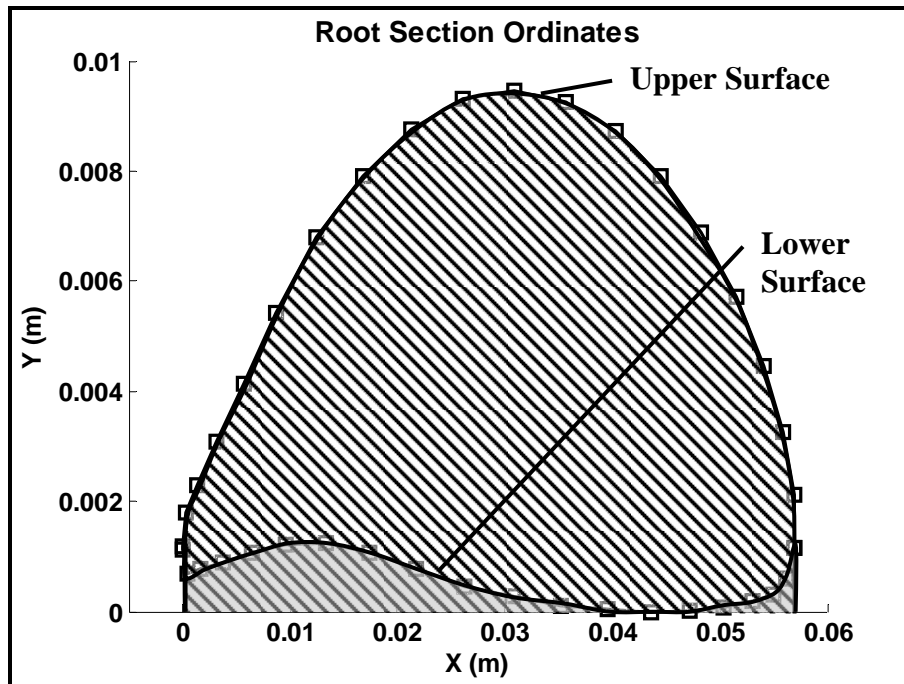


Figure 15: Distorted Root Section

Figure 15 shows the visually distorted root section of the propeller shown in Figure 14. The section is distorted for illustrative purposes; the undistorted root section is shown in Figure 16.

OpenProp treats the blade section as an upper and lower surface. The overall procedure for determining blade section area properties consisted of determining the area properties of the area formed by the upper surface and the x-axis and subtracting the area properties formed by the lower surface and the x-axis. This subtraction results in the properties of the enclosed area shown above.

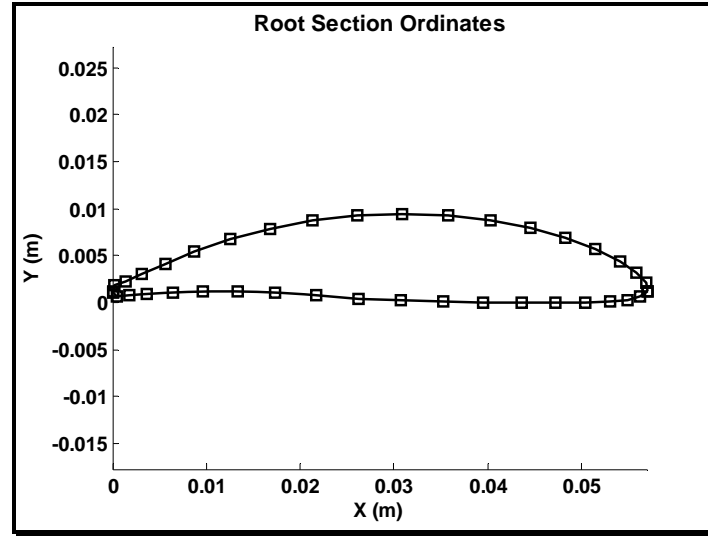


Figure 16: Undistorted Root Section

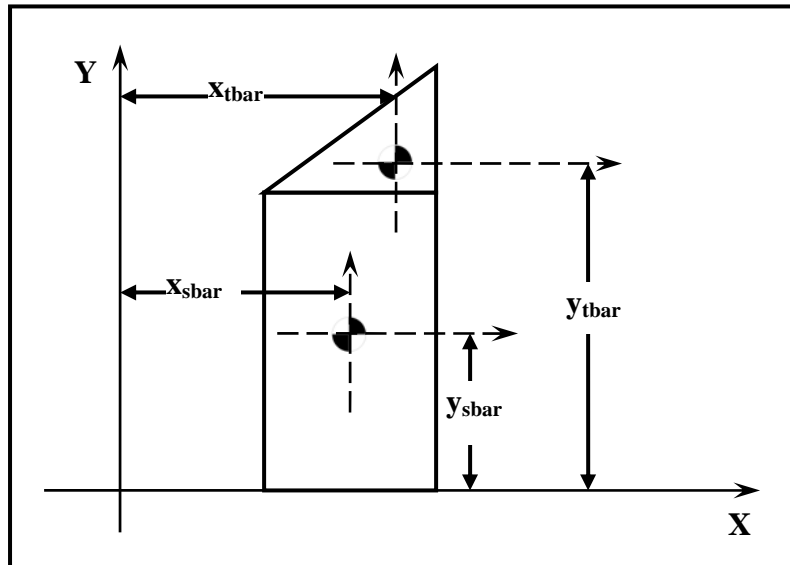


Figure 17: Calculation of Elemental Area Properties

Figure 17 shows a characteristic diagram that was used to determine elemental area properties which were summed to achieve the section area properties. The procedure was:

1. Determine elemental area
2. Calculate elemental centroid
3. Calculate elemental 2nd moment of area about both x and y axes
4. Sum elemental areas
5. Sum 2nd moment of areas about x and y axes
6. Calculate section centroid, Equation 13

$$\bar{Y} = \frac{\sum \bar{y}_i A_i}{\sum A_i}, \bar{X} = \frac{\sum \bar{x}_i A_i}{\sum A_i} \quad (13)$$

7. Apply parallel axis theorem to determine 2nd moment of area about the centroidal axes, Equation 14.

$$I_{Centroid X} = I_X - \bar{Y}^2 A_{total}, I_{Centroid Y} = I_Y - \bar{X}^2 A_{total} \quad (14)$$

In order to determine the other quantities required by Equation 12, the integrals were turned into discrete sums and variables from the propeller design were used.

Results

The results of the analysis performed for the propeller described in Epps (2010a) are shown below for an on-design and off-design condition. Figure 18 shows the stress at the blade root. As expected, the blade is in tension on the pressure side and compression on the suction side. Note that the stresses indicated in Figure 18 in the middle of the root section are interpolated stress. Only the stresses at the blade surface were calculated at the points indicated.

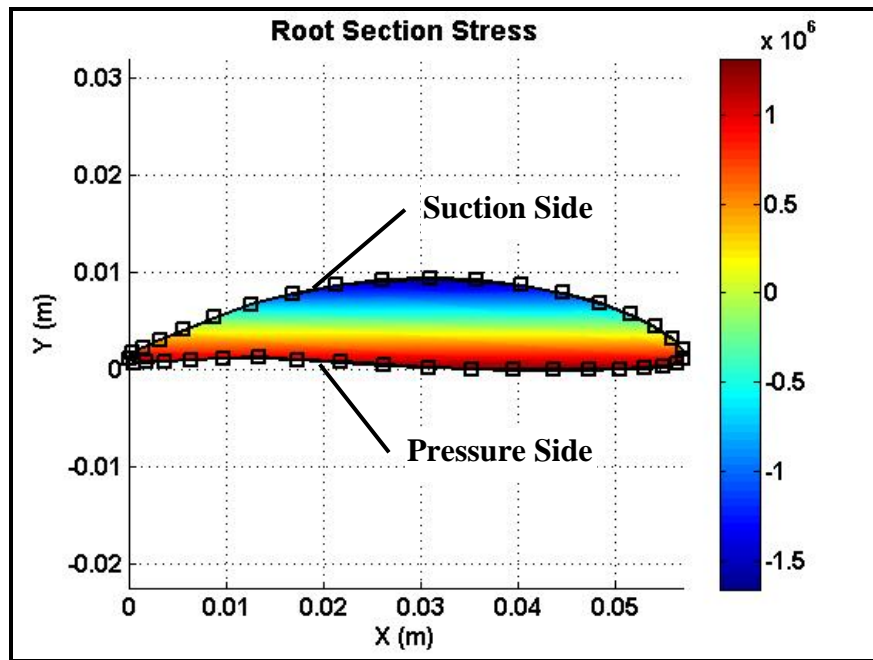


Figure 18: On-design Root Section Stress

In Figure 19 through Figure 22 tensile stresses are considered positive and compressive stresses negative. Figure 19 and Figure 20 represent the on-design condition while Figure 21 and Figure 22 represent an off-design condition as specified in the figure titles. As expected, the off-design condition chosen shows higher stresses than the on-design condition because the off-design condition corresponds to a point where the propeller is producing greater thrust and torque. Greater thrust and torque results in higher stress.

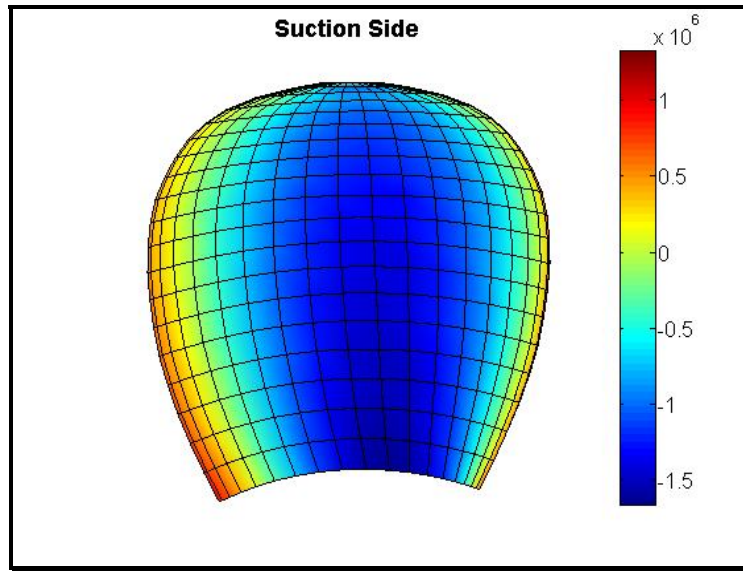


Figure 19: On-design Suction Side Stress: $J_s=0.75$, $V_s=1.5m/s$, $n=8rev/s$, $D=0.25m$

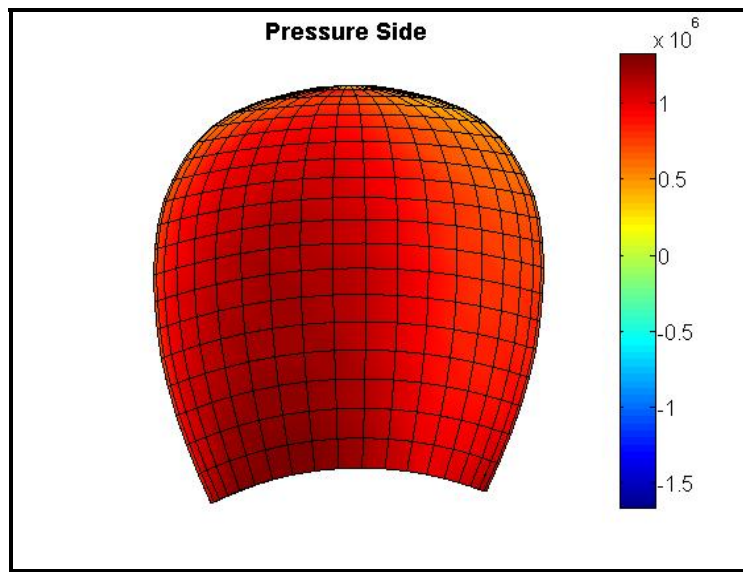


Figure 20: On-design Pressure Side Stress: $J_s=0.75$, $V_s=1.5m/s$, $n=8rev/s$, $D=0.25m$

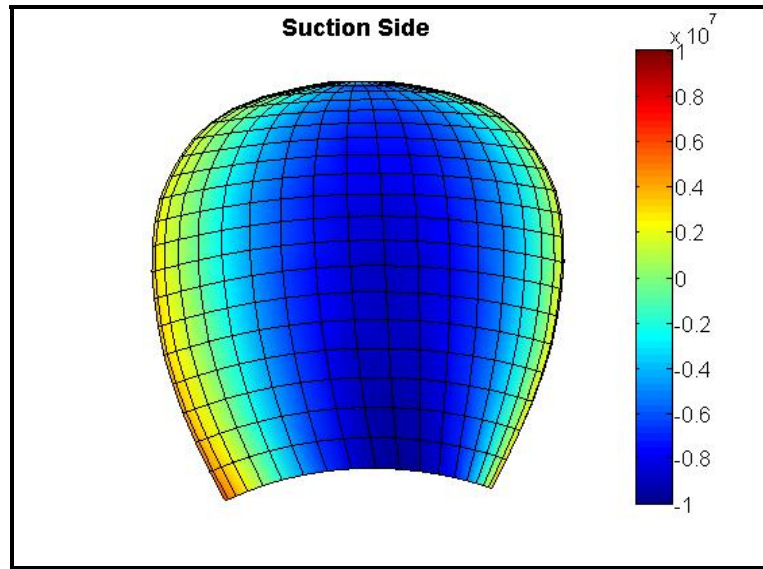


Figure 21: Off-design Suction Side Stress: $J_s=0.40$, $V_s=1.5m/s$, $n=15rev/s$, $D=0.25m$

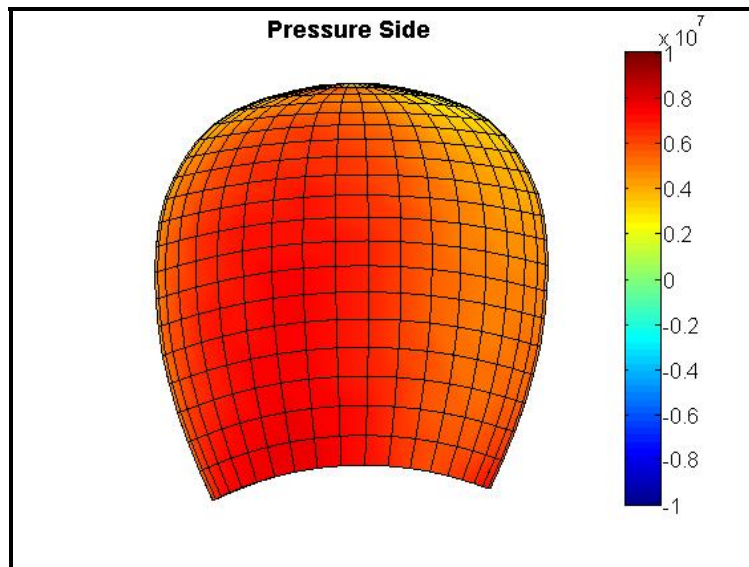


Figure 22: Off-design Pressure Side Stress: $J_s=0.40$, $V_s=1.5m/s$, $n=15rev/s$, $D=0.25m$

Carlton (2007) presents isostress contour lines taken from Finite Element Analysis (FEA) results for various propeller types. The results presented above agree with the trends presented by Carlton for a propeller blade without skew. Carlton shows highest stress near the blade mid-chord in a region that extends close to the tip of the blade and a decreasing stress as one moves away from the mid-chord to the blade leading and trailing edges.

Limitations

There are several assumptions and simplifications that are made in the method discussed above. Each of these is listed below with suggestions to improve the calculation.

1. Blade Section – In propeller design it is customary to present the blade section geometry as the unwrapped section. In other words, the blade section geometry is the geometry one would obtain if the curved blade section were laid flat. This geometry was used in the calculation of the blade section properties. It is more desirable to calculate the blade section properties for blade sections that were taken using a flat cutting plane oriented perpendicular to the span line. The implementation of stress calculations would be more difficult for a truly flat blade section since OpenProp does not develop the blade geometry for this type of section nor is the hydrodynamic data valid for a blade section obtained in this way. The OpenProp geometry code could be rewritten to perform an interpolation in order to obtain the points to create a truly flat blade section but the problem of obtaining blade section loading remains. Blade loading could possibly be obtained using the cavitation analysis module pressures however the stress code would then need to be altered to be something similar to FEA.
2. Geometric Property Calculation – The method used to calculate blade section properties was essentially a trapezoid rule integration. This method was used because it does not limit the selection of the number of points used to design the blade, however other more precise methods could be used.
3. Point of Load Application – In the method presented, the section lift force is applied to the section centroid. This simplifies the code because the actual point of application does not have to be determined and the resulting torsional loads can be ignored. The actual point of lift force application could be approximated as the $\frac{1}{4}$ chord point or could be calculated more precisely by analyzing the pressure distribution from the cavitation analysis module. If the load point is corrected, the resulting torsional shear stresses could be calculated. This correction is probably small for blades without skew.
4. Shear Stresses – All shear stresses on the blade section are ignored. The stresses calculated above represent bending stress from lift and axial stress from the centrifugal force. Shear stress could be included if the load point is corrected and the resulting torsional stress is calculated and added to the shear stress resulting from the pressure differential between the blade faces. If shear stresses are included, the principal stresses should be calculated.

Chapter 6 – Fatigue Analysis

By definition fatigue failure is characterized by a time varying load whose magnitude is smaller than that required to produce failure in a single application, Pook (2007). The fatigue analysis conducted as part of this project is presented in two steps.

1. Identification of the cyclic loading
2. Application of a fatigue failure theory.

A comprehensive fatigue analysis is characterized by many subtleties and in many cases significant experience is necessary to conduct the art of a fatigue analysis. The fatigue analysis presented here is intended to provide a method by which a fatigue analysis could be conducted on a propeller or turbine at the beginning of the design process to ensure the estimated fatigue life meets the design goal. As a whole, OpenProp is intended to be a design tool which can be used to provide good initial propeller and turbine designs. As additional iterations of the design process are completed more sophisticated tools for propeller design will become necessary. It is in this spirit of providing good initial design estimates that the fatigue analysis is presented here.

Cyclic Load

For a propeller or turbine the source of the varying load is the wake that it operates in. Due to the presence of a wake, the inflow velocities to the blades are not uniform in magnitude or direction. As a blade completes a revolution it will pass through regions of various velocity which will induce varying forces on the blade. A propeller will typically operate in a wake with greater inflow velocity variation than a turbine. Because a propeller operates in a more severe wake environment and because wake data is more readily available for propellers than for turbines, fatigue analysis for a propeller was performed.

A wake for a single screw ship is shown in Figure 23. This wake is also shown in Laskos (2010) and was measured by Koronowicz, Chaja and Szantyr (2005). This figure clearly shows a circumferential variation in the axial inflow velocity. Typically there is also a circumferential variation in the tangential inflow velocity but this variation is much smaller and is not considered here. This is shown in wake profiles of Felli and Felice (2005). Figure 23 shows the ship wake divided into twelve sectors. As the blade passes through each sector it is assumed to fully develop lift commensurate with the flow velocity in that sector. This assumption makes this analysis a quasi-steady analysis. In each sector, the circumferential average of the axial inflow velocities was taken at the same radial positions that were used in the propeller design. Each blade section is subjected to a different inflow velocity which results in a different C_L . In order to determine the new C_L on each section Equation 15 was used.

$$C_{Lf} = C_{Lo} + 2\pi(\Delta\alpha) \quad (15)$$

where

C_{Lo} = original lift coefficient in the design condition

C_{Lf} = new lift coefficient at the new angle of attack

$\Delta\alpha$ = change in angle of attack from design condition

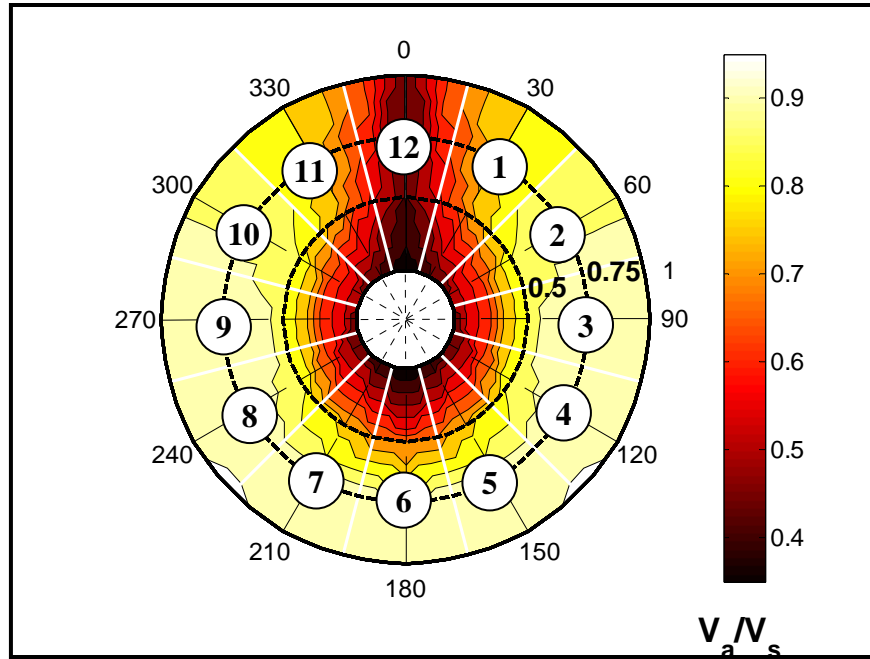


Figure 23: Sectored, Single Screw Ship Wake

Figure 24 and Figure 25 show how a change in the axial velocity produces a change in the magnitude and direction of the total inflow velocity. The analysis also assumes that u_a^* and u_t^* remain constant.

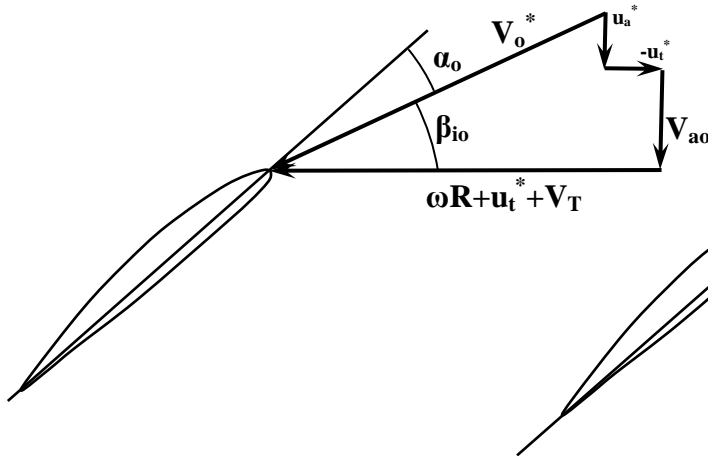


Figure 24: Original Inflow Velocities

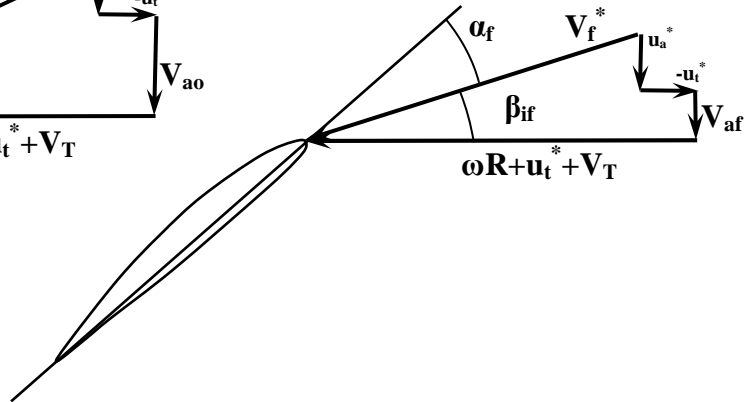


Figure 25: New Inflow Velocities

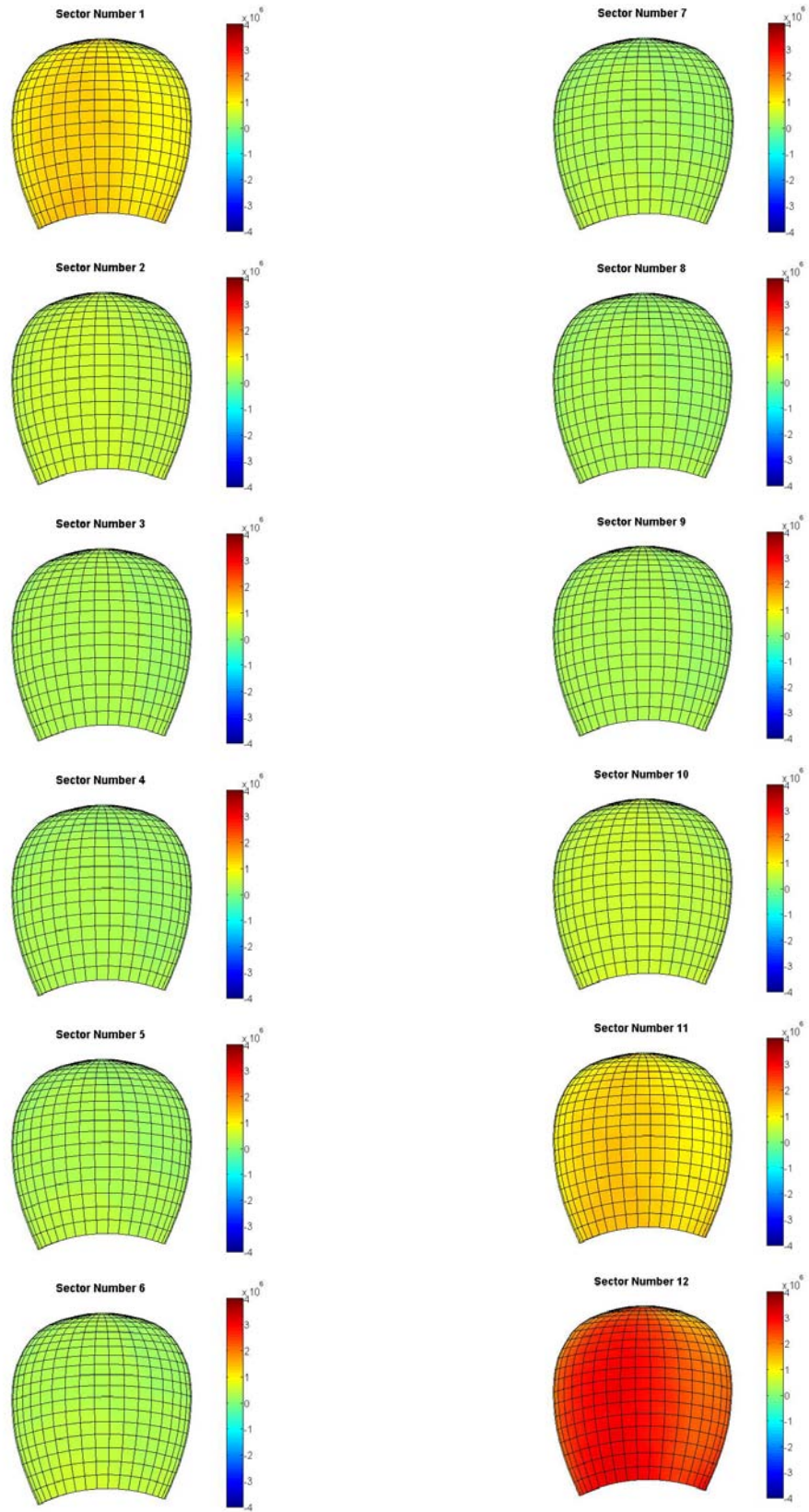


Figure 26: Pressure Side Blade Stress for Each Wake Sector

Figure 26 above shows the change in blade stress as the blade passes through the wake sectors of Figure 23. As expected, the highest stresses occur in sector number twelve where the axial inflow velocity is the lowest. The lowest axial inflow produces the largest angle of attack and lift coefficient and subjects the blade to the largest amounts of lift and stress.

Since the blade stress varies considerably across the blade faces, it is necessary to identify the point where maximum tensile stress occurs. The point of maximum stress for this propeller occurs at the blade root at the point identified by the arrow in Figure 27.

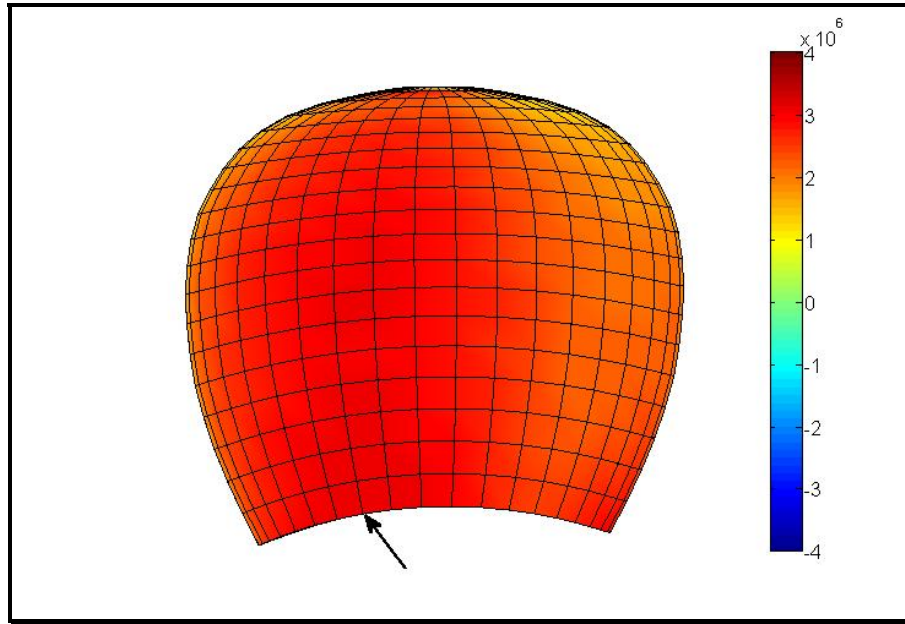


Figure 27: Point of Maximum Tensile Stress

In Figure 28, plots of the maximum blade stress versus angular blade position for various ship speeds are shown. These plots also identify the alternating stress, σ_a , associated with each blade stress. Except for the highest ship speeds, σ_a is relatively low, near the endurance limit for nickel, aluminum bronze, as shown in Figure 29.

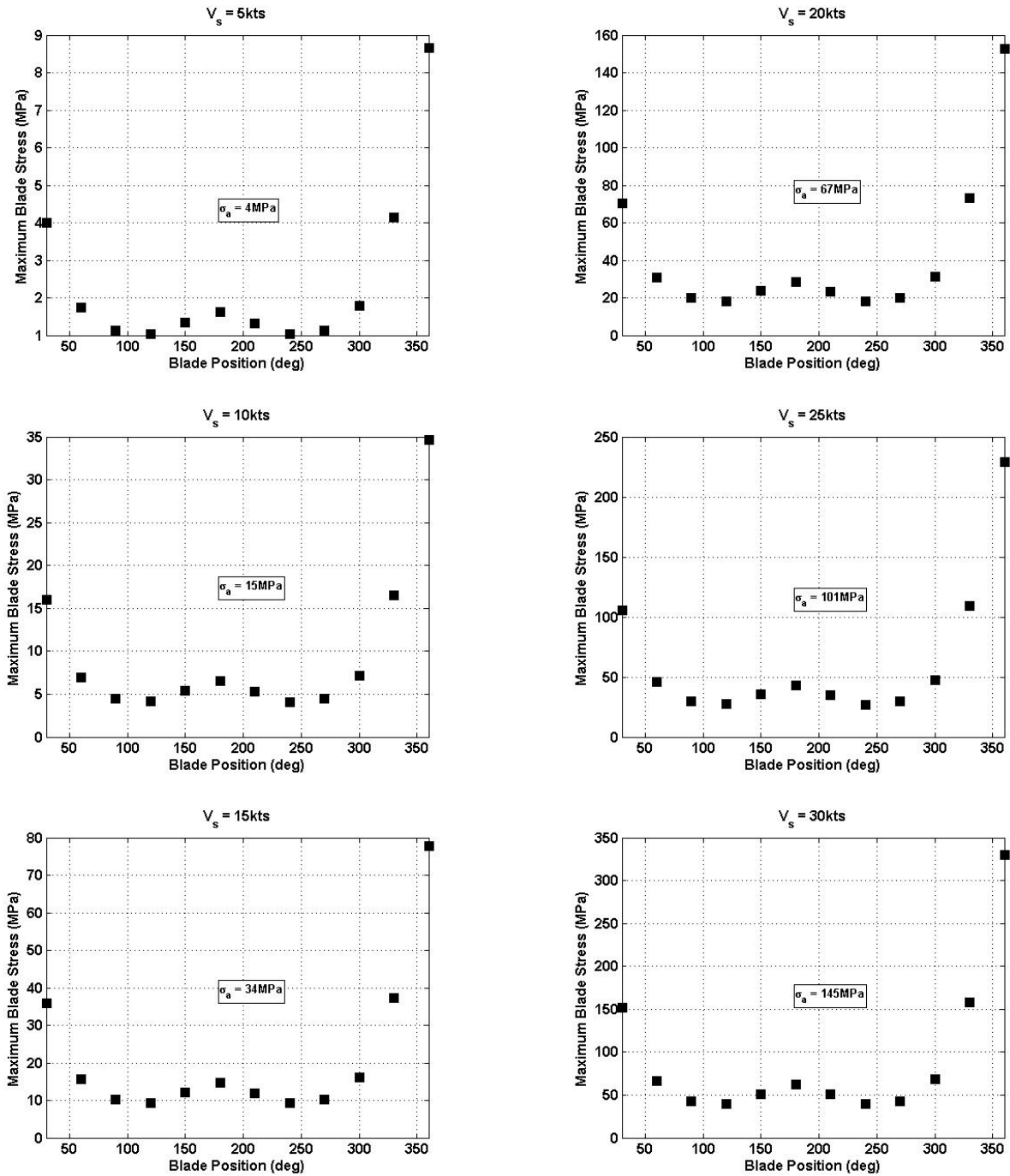


Figure 28: Maximum Blade Stress versus Angular Position for Various Ship Speeds Using the On-Design Advance Coefficient ($J_s=0.75$)

Fatigue Failure

Figure 29 shows a plot of σ_a , versus number of reversals/cycles to failure for nickel, aluminum bronze. Data for this figure was taken from Kerwin and Hadler (2010); detailed alloy composition and test condition are unknown. Ideally, one would design a propeller such that blade stresses were minimized in order to increase the fatigue life of the propeller.

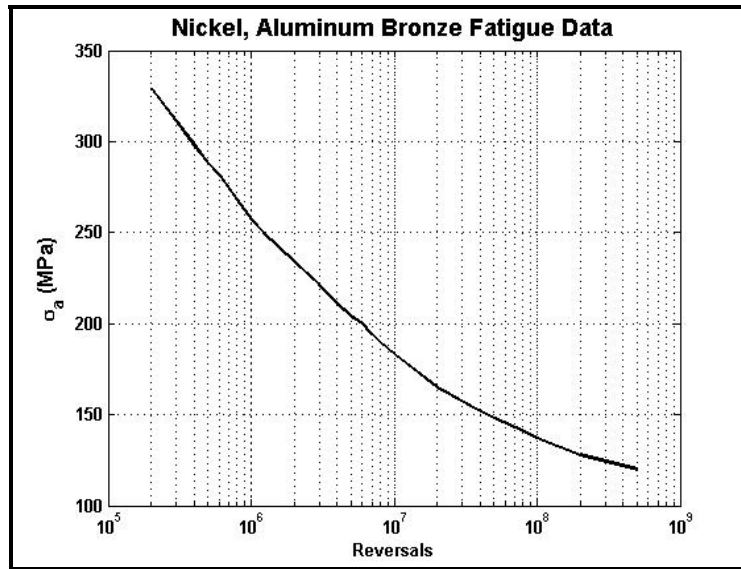


Figure 29: S-N Curve for NiAl Bronze

When performing a propeller fatigue analysis it is critical that the operational profile of the ship is taken into consideration. Figure 30 shows an operational profile for a warship which was taken from Gooding (2009). Since the propeller analyzed here was not analyzed for such a wide spectrum of speeds, Figure 31 was used in the example calculation.

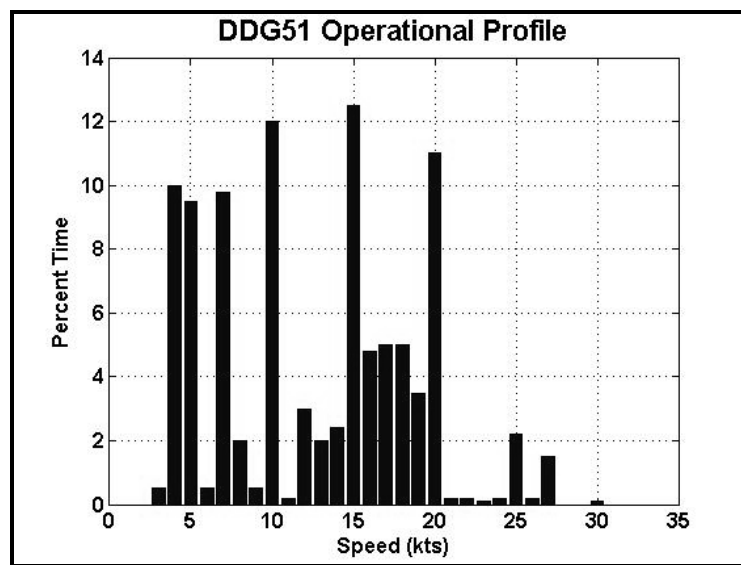


Figure 30: Operational Profile for DDG51

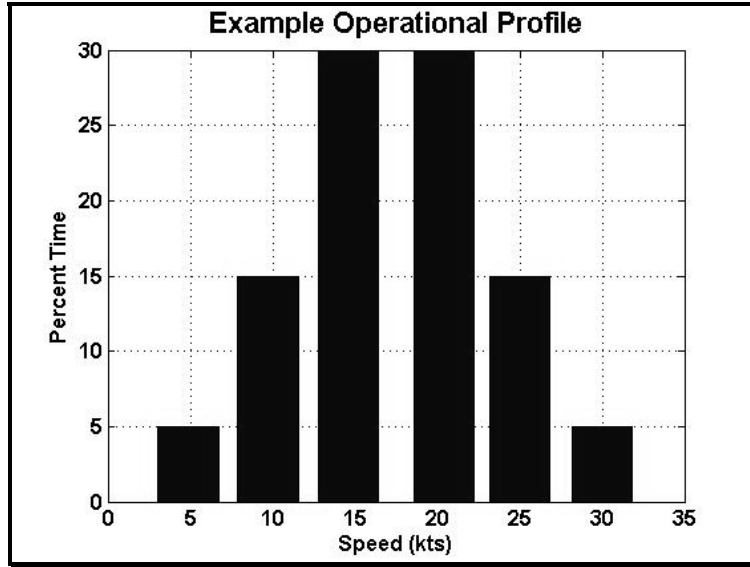


Figure 31: Example Operational Profile Used for Calculations

With the assumptions made in this analysis, there is a direct correlation between ship speed and blade stress. This correlation was used to produce Figure 32 below.

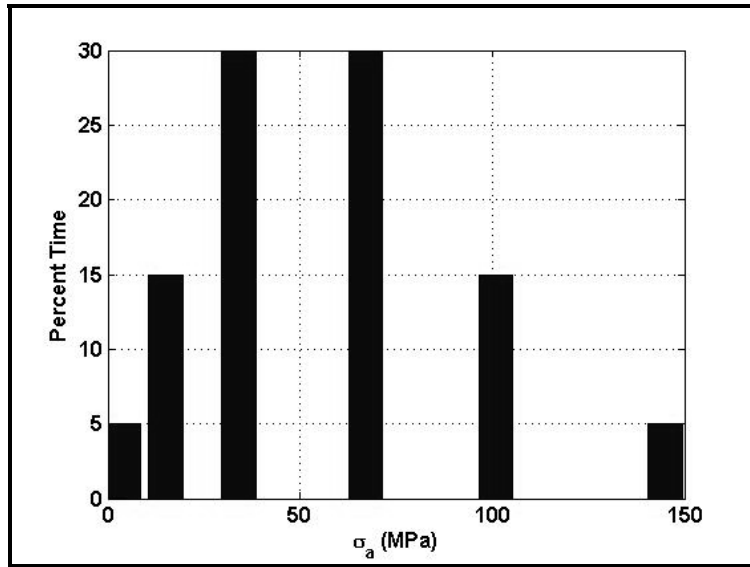


Figure 32: Time at Various Stress Levels

Miner's rule was used to predict the fatigue life of the propeller. Miner's rule is simply stated in Equation 16.

$$\sum \frac{r_i}{R_i} = 1 \quad (16)$$

where

r_i = actual number of reversals at σ_a

R_i = reversals to failure at σ_a , determined from Figure 29.

In order to predict the fatigue life, additional equations are necessary. These are shown below.

$$r_i = RPM_i t_i \quad (17)$$

where

t_i = time spent at rotation rate, RPM_i

RPM_i = rotation rate which produces desired speed

$$t_i = x_i T \quad (18)$$

where

x_i = fraction of total time spent at RPM_i

T = total time of propeller operation

Substituting Equation 17 and Equation 18 into Equation 16 and solving for T , one obtains:

$$T = \frac{1}{\sum \frac{RPM_i x_i}{R_i}} \quad (19)$$

If one considers the blade stress at speeds below 25kts to be of infinite life then the fatigue life is 180 days. This calculation is dominated by the time spent at 30kts which is probably excessive when comparing Figure 31 and Figure 30.

Limitations

In addition to the limitations discussed at the end of Chapter 5, there are some additional limitations that are specific to the fatigue analysis. While the method presented here is useful to obtain an initial estimate of fatigue life there are many more factors which should be considered as the initial estimate is refined.

1. **Fatigue Data** – Figure 29 shows notional fatigue data for a Ni-Al Bronze. It is unknown how this data was obtained and to what specific alloy it applies. It would be desirable to use data obtained in a seawater environment for the specific alloy one intended to use to manufacture a propeller.
2. **Miner's Rule Coefficient** – In the method above, it was assumed that when the sum of Equation 16 reached unity that the material would fail. This is a reasonable assumption but others have found that the coefficient should be a number other than one depending on the specific material type (Sines and Waisman 1959). Determination of a more accurate value for this coefficient would involve an extensive test program that should be performed for more refined calculations.
3. **Shaft Reversals** – Shaft reversals are a routine operation in ship maneuvering, particularly when the ship is near a pier, in high traffic areas or navigating waters that restrict its turning ability. Kerwin and Hadler (2010) and Carlton (2007) provide a limited discussion on the effect of shaft reversals on blade fatigue life. A shaft reversal can be a highly stressing event for the propeller blade and therefore can significantly reduce the blade fatigue life. Calculating the blade loads during a shaft reversal would require unsteady analysis and is beyond the scope of this project and the current capabilities of OpenProp.
4. **Blade Response** – The analysis presented here assumes that the blade will completely respond to the flow regime of each sector. It is unlikely that this is the case. The analysis does not take into account the time element necessary for the blade to fully develop its thrust and in this regard is a conservative estimate of fatigue life.

Chapter 7 – Test Fixture Design and Construction

This chapter describes the design and construction of a test fixture for testing propellers and turbines. The test fixture described in this chapter was specifically designed for use in the hydrodynamics laboratory water tunnel at MIT but can also be used in a tow tank. The limitations of the test fixture are given in the Table 4.

Limits	Value	Basis
Torque	6 ft-lbf	Sensor limitation
Thrust	50 lbf	Sensor limitation
RPM	1500 rpm	Peak capability of motor
Current	18 amps	Peak capability of motor
Voltage	240 V-AC 300 V-DC	Required supply voltage Maximum controller output voltage

Table 4: Test Fixture Limitations

The design philosophy employed for this test fixture, with accompanying justification is given below.

1. Thrust and torque sensor must be the limiting component. The sensor used in this test fixture is on loan to Professor Richard Kimball from the US Navy. Searches for a commercially available sensor capable of simultaneous thrust and torque measurement did not yield any devices that could have been used in a test fixture of this size. Because of the limited availability of useable sensors, it was decided that the test fixture should be limited only by the sensor.
2. Components must be usable in other test fixtures. Since there were no other test fixtures of this type at MIT, this design constraint meant that the test fixture must be able to be disassembled and the components able to be used in other test fixture assemblies that might be designed by students in the future. This constraint was a significant driver in the selection of electrical components, manufacture of mechanical components and method of component assembly.
3. Fixture must be able to incorporate a high resolution encoder. This constraint effected the motor and encoder selection process.
4. Fixture must be capable of use in both a tow tank and water tunnel. This constraint drives the maximum allowable overall diameter, length and standpipe length of the test fixture.

Additional details concerning how the design philosophy impacted test fixture design as well as the final test fixture configuration are given in the sections that follow.

MECHANICAL

Thrust/Torque Sensor

The thrust/torque sensor used in this test fixture is a strain gage type sensor. The sensor uses two sets of strain gages; one set to measure thrust and the other set to measure torque. The strain gages are adhered to the center ring shown in Figure 34, which is covered in an opaque epoxy like material. The presence of this material introduced measurement error when building the

CAD sensor model that was created and is one of the reasons why a factor of safety (FOS) of 2 was used when determining the maximum operational torque that could be applied to the sensor.

Since the sensor was to be the limiting component, it was necessary to characterize the thrust and torque capabilities of the sensor. In order to determine the maximum thrust and torque that the sensor could measure without damage, a determination of sensor material was made and FEA of the sensor was performed. For the purposes of this test fixture design, “damage” is defined as a load condition which would produce yielding in the sensor material. FEA required that a three dimensional model of the sensor be made, this model is shown in Figure 33.

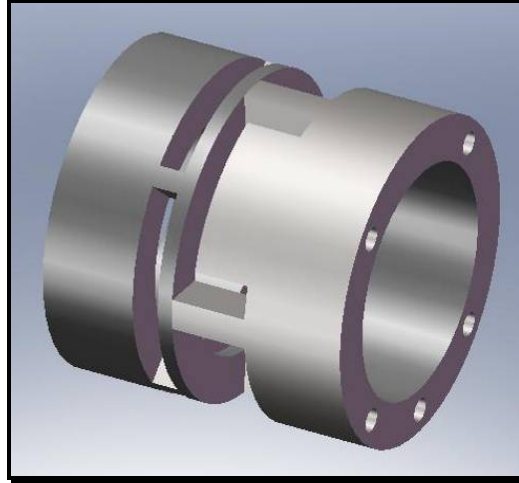


Figure 33: Provided Sensor

While measuring the sensor to determine physical dimensions for incorporation into the model, an inscription of 50lbf was found on one end of the sensor. 50lbf was used as the thrust load on the sensor in the FEA analysis in order to determine a FOS. The result of the FEA showed that the sensor can withstand a 50lbf axial load with a FOS of 2. The calculated stress distribution resulting from a 50lbf load is shown in Figure 34.

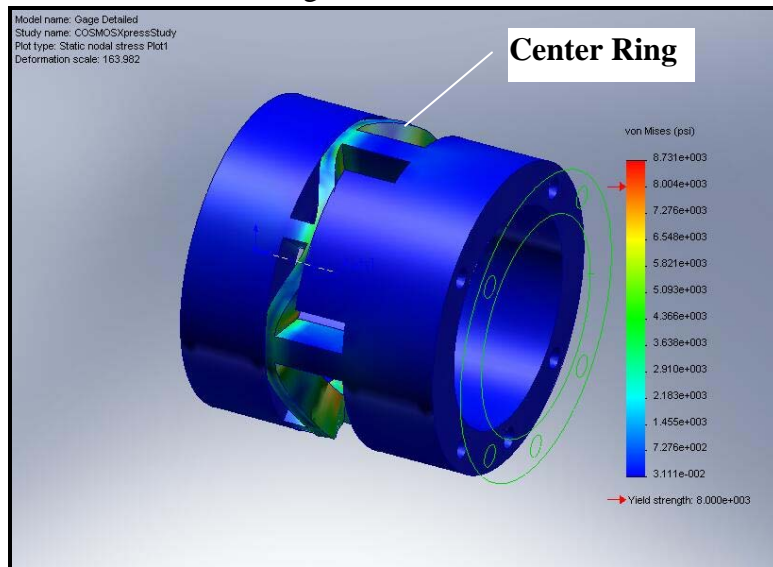


Figure 34: Stress from Axial Load on Sensor (50lbf applied)

In order to determine the maximum torque that the sensor could carry, a separate FEA was conducted. The results of this analysis show that the sensor could carry 12ft-lbf without damage. Application of a FOS of 2, that was determined from the thrust FEA, limited the maximum torque of the sensor to 6ft-lbf. A FOS of 2 is reasonable due to the dimensional error present in the model and a lack of validation of the FEA used on the model of the sensor. A picture of the stress distribution resulting from a 12ft-lbf applied torque is shown in Figure 35. Note that the “handle” that is present in the picture was necessary to apply a torque load in SolidWorks® 2007 Education Edition.



Three options were considered for the configuration of the output shaft to which a propeller or turbine could be attached for testing.

- Option 1 was undesirable because the shaft size required to accommodate the taper would have required larger bearings and seals for the shaft which would have increased the friction resistance on the shaft and made sealing the shaft more difficult. Additionally, a larger diameter shaft has greater rotational inertia which would limit the rate at which the shaft could be accelerated during unsteady tests.

Option 3 was less desirable than Option 2 because of the complication of manufacturing propellers with a set screw hole. The intended manufacturing technique for propellers is 3D printing. Propellers manufactured using this method are made from ABS plastic. Successfully creating a threaded hole into this material with sufficient holding power for a set screw seemed unlikely. A second problem with this type of shaft is that it required a female section to be made in the propeller hub that would have been difficult to machine: a straight cylindrical hole that changes to a cylindrical hole with a flat. Previous experience manufacturing propellers using the 3D printing technique has shown that it is difficult to achieve a hub whose outer diameter is concentric to the drive shaft hole outer diameter. Therefore it is necessary to turn the propeller on a lathe to ensure that the drive shaft will easily attach to the propeller with minimal eccentricity between the inner and outer diameter of the propeller hub.

Option 2 requires that every propeller have a slot machined in the hub but this operation is simple using an end mill of the same size as the output shaft pin. It is also possible to print the slot in the hub if the turbine is manufactured using a rapid prototyping technique. Option 2 also requires that the end of the drive shaft be threaded to accept a nut to hold the propeller against the drive shaft pin, however these are external threads that are easy to manufacture. For these reasons, Option 2 for drive shaft configuration was chosen. A picture of the shaft is given Figure 36.

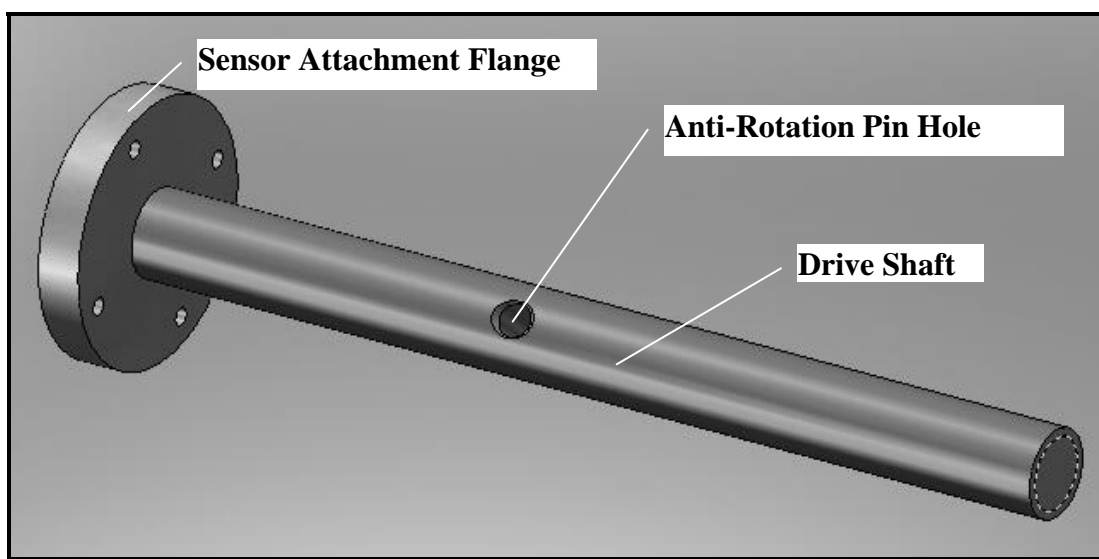


Figure 36: Output Shaft Configuration

Drive Shaft Configuration

The test fixture design described in this paper is intended to be used to test both propellers and turbines. Because of this dual use capability, it is necessary that the fixture be able to measure and support axial loads in two directions. Including the capability to support axial loads in two directions also protects the fixture from inadvertent damage should a load be applied in an axial direction for which the fixture was not designed.

Dual axial load support was accomplished by using two tapered roller bearings in an arrangement similar to the front wheel bearing assembly on an older automobile. The tapered roller bearings are mounted in a bearing assembly in such a way that one bearing supports the axial load in one direction and the second bearing supports the axial load in the other direction. The drive shaft in the vicinity of these bearings is threaded and slotted to accommodate an axle nut and star washer. The nut ensures the bearings are secured in the bearing housing and that the axial play in the drive shaft can be adjusted. The slot in the shaft, in combination with the star washer, ensures that the nut will not loosen. A picture of the drive shaft and bearing assembly is shown in Figure 37.

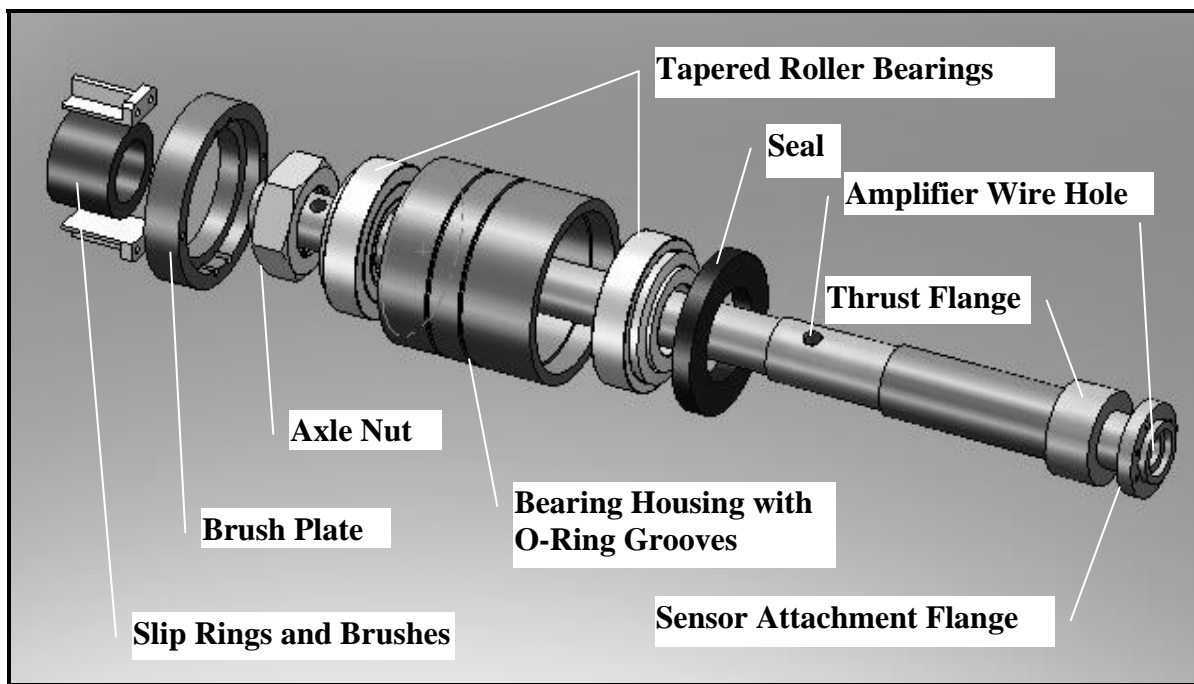


Figure 37: Driveshaft and Bearing Assembly with Brush Blocks and Slip Rings

The smallest diameter on the driveshaft was determined by the diameter of the slip ring assembly. Due to the shoulder required for the tapered roller bearings, the slip ring assembly can only be installed from one end of the shaft. The end of the shaft over which the slip rings must be moved to reach the installation location was made slightly smaller than the slip ring diameter in order to ease slip ring installation. The drive shaft diameter for installation of the slip rings is only slightly smaller than the shaft diameter required for the tapered roller bearings. This small change in diameter meant that little material was available to make the threads for the axle nut and therefore a custom nut, washer and thread configuration had to be manufactured.

Housings

Two assembly methods were considered for the external housings.

1. Threaded assembly
2. Shoulder fasteners

Using a threaded assembly has the advantage of minimizing the number of water leakage paths into the fixture and the number of o-rings required during assembly. The problem with a threaded assembly is that the threads can be difficult to manufacture, particularly for internal threads that run deep into the part, and large diameter threads are prone to seize in stainless steel. The problem with the shoulder fastener assembly method is that the number of leakage paths and o-rings required is significant and assembly requires that the components be precisely positioned prior to the installation of the shoulder fasteners. The shoulder fastener assembly method was chosen for ease of manufacture and the problem of water leakage paths was mitigated by installing the shoulder fasteners between a set of o-rings on the housing diameters.

ELECTRICAL

Slip Rings

The torque/thrust sensor uses two sets of strain gages for load measurement. There are several possible methods to transmit the data signal from the sensor for recording. The method chosen for this fixture was to amplify the data signal at the sensor and then use a set of slip rings and brushes to conduct this signal to a point where a data acquisition system could be attached. This method was chosen for its simplicity.

The details of the slip ring assembly are included in the appendix. Six slip rings are required for sensor operation. Two are necessary to power both sets of strain gages, four rings are necessary for data signal transmission. The slip assembly used in this test fixture has eight slips rings in order to allow for future growth and to provide alternate slip rings should some become unusable. Each slip ring has four brushes riding on it, two from each brush block. The brushes from each brush block are soldered together so that four brushes are connected to each slip ring. Four brushes per slip ring are used in order to minimize the electrical resistance between the brushes and the slip rings. A photograph of the slip rings and brushes installed in the test fixture is shown in Figure 38.

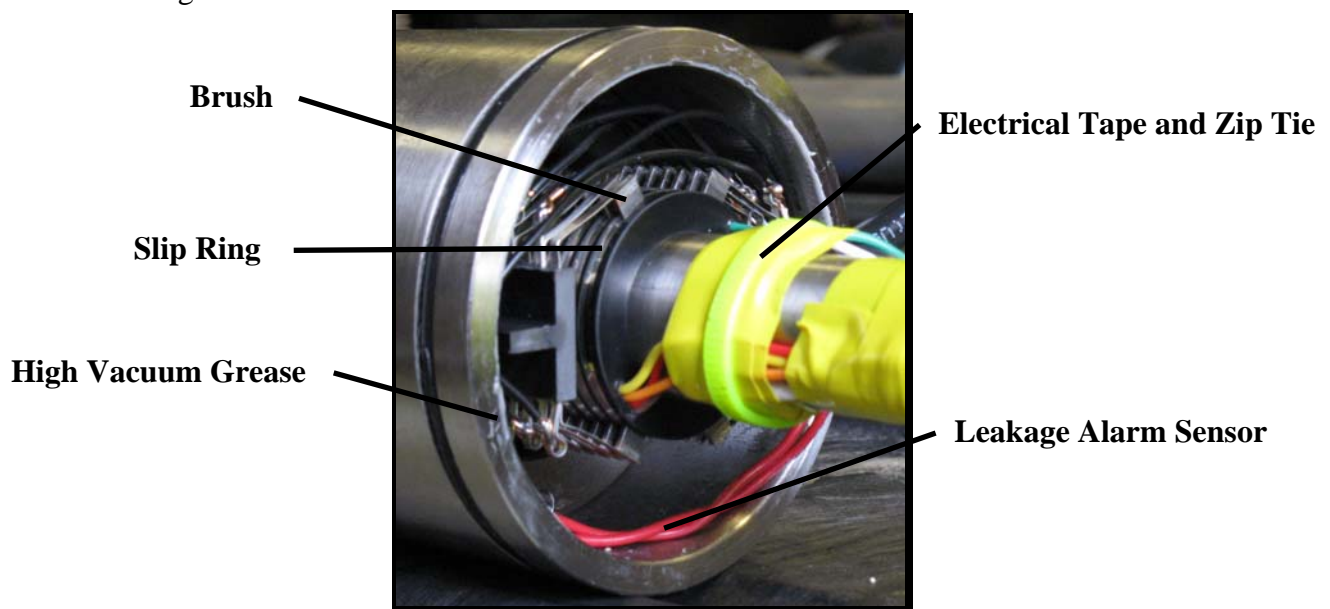


Figure 38: Installed Slip Rings and Brushes

In addition to the slip rings and brushes, Figure 38 shows two red wires near the bottom of the photograph. These wires are part of a leakage alarm system. If the ends of the two red wires are shorted, an audible alarm sounds indicating leakage into the test fixture. The yellow electrical tape and zip tie are present to secure the slip ring and strain gage amplifier wiring to the driveshaft. The connection of the slip ring wires and the strain gage amplifiers was made outside of the hollow drive shaft in order to ease assembly. The white substance on the end of the tube is a high vacuum silicone based grease that is applied to the surfaces prior to assembly in order to ease assembly and as an additional measure to prevent leakage.

Amplifiers

Inside the thrust/torque sensor are two amplifiers. One amplifier is for the thrust data signal and the other is for the torque data signal. These amplifiers are mounted inside a piece of foam which is pressed into the sensor. The amplifiers that were purchased are designed for strain gage signal amplification for the motor sports industry and therefore represent a rugged option for signal amplification. Data signal amplification takes place as close to the sensor as possible in order to limit the data signal transmission loss and to prevent the signal to noise ratio of the data signal from becoming too low for practical use. Additional amplifier details are included in the appendix.

Motor

The desire to use the test fixture in the water tunnel limits the maximum allowable diameter of the test fixture. Previous experience with trolling motors in the water tunnel yielded good results. Trolling motor diameters ranged from 3.5 to 4 inches; therefore the maximum allowable test fixture diameter was set to 4 inches. A maximum diameter of 4 inches significantly restricts the available options for motor selection.

Another consideration in motor selection is the ability of the motor to also act as a generator in order to serve as a load for turbine testing. The requirement to also act as a generator further limits the choice of motor to those of a permanent magnet design. Although it is possible to use a motor without permanent magnets installed, the complication arising from supplying both the stator and rotor with electric current was deemed excessive for a test fixture.

The motor selected for this test fixture is a Parker kit motor, K089300. This motor is a DC brushless motor; the specific model selected also contains integral commutation. In selecting a motor, it was desirable to select a motor such that the sensor remained the limiting component in the design. Therefore a motor capable of torque in excess of 16N-m (12ft-lbf) was selected. A test fixture using a standard motor required a test fixture 6 inches in diameter.

A standard motor with the desired torque speed characteristics exceeded the maximum allowable diameter because a standard motor comes with a face plate on one end and electrical connectors on the other. The standard motor would have required customization to remove the electrical connectors and change the mounting configuration to a face frame mount. In addition to the complication and cost of performing the customization, supplying the motor with current would

have also been challenging because the wires would have had to pass by/through the face plate area in order to be routed to the standpipe for passage out of the test fixture to the electrical supply.

The K089 series motor has a maximum diameter of 3.5 inches which allows the motor to be attached inside a tube with a maximum outside diameter of 4 inches. The K089300 was selected because it is the highest torque motor listed in the catalogue for this series. Parker frameless kit motors generate additional torque in a given series by increasing the length of the stator windings and rotor. A kit motor has the following advantages:

1. Able to fit in a tighter package
2. Use of one shaft for both the motor and driveshaft which eliminates the need for a shaft coupling between motor shaft and propeller drive shaft
3. Stator windings are in direct contact with the test fixture housing which allows efficient heat transfer out of the stator windings and into the fluid surrounding the test fixture.

The disadvantages of using a kit motor are:

1. Attachment of the motor into the fixture required that additional holes had to be drilled into the motor housing which meant an increase in the probability of a leak into the fixture.
2. Kit motors do not come with a high resolution angle encoder installed like the standard off the shelf motors.

The calculated torque speed curve for the K089300 is given in Figure 39. In both figures the dashed lines represent continuous operation while the solid lines represent peak or intermittent operation. The linear negative slope in the torque-speed curve is based on preventing the motor windings from overheating due to excess current.

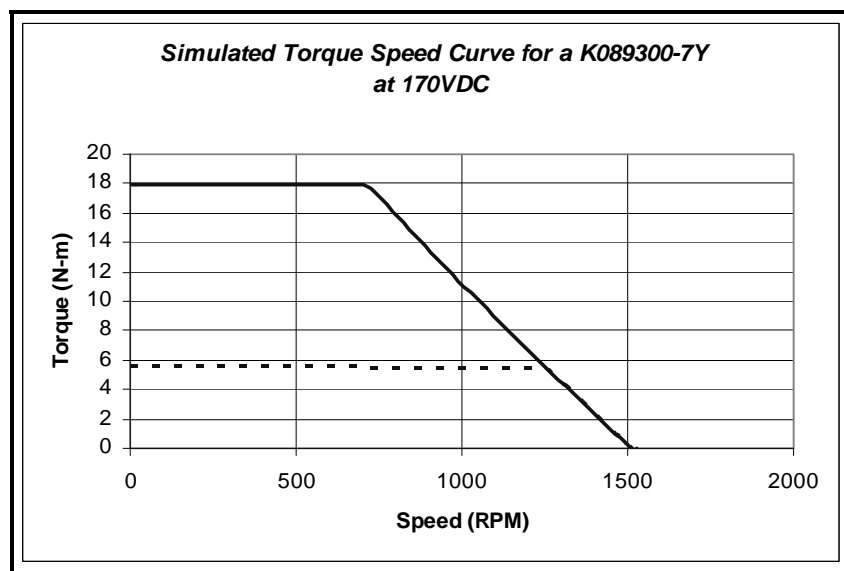


Figure 39: K089300-7Y Torque Speed Curve

0.

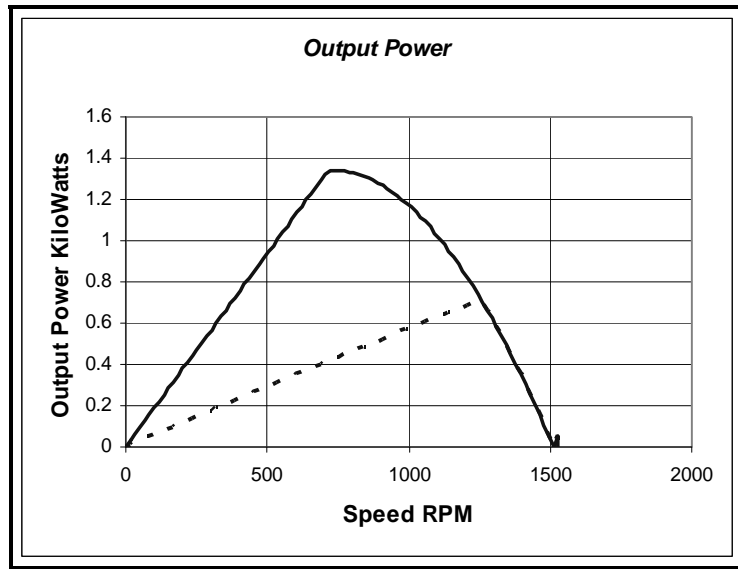


Figure 40: Output Power Capability

The current limitation of the motor is shown in Figure 41.

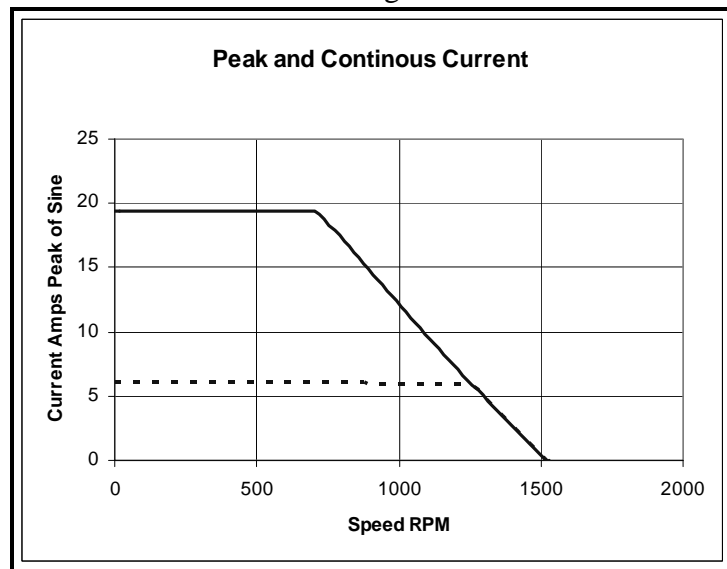


Figure 41: Limiting Current

Controller

In selecting a controller it was desirable to select a controller which could serve as both a motor controller and load controller, allowed for test fixture growth and had limited EMI to prevent noise in the data signal. Students at the University of Maine have built and used a test fixture for cross flow turbines that used a Copley Xenus XTL-230-40 controller. They have been very pleased with the overall performance of their test fixture, particularly the low electrical noise

generated by the controller. For these reasons the same controller and electrical layout were selected for this test fixture. A photograph of the electrical components in the enclosure is shown in Figure 42.

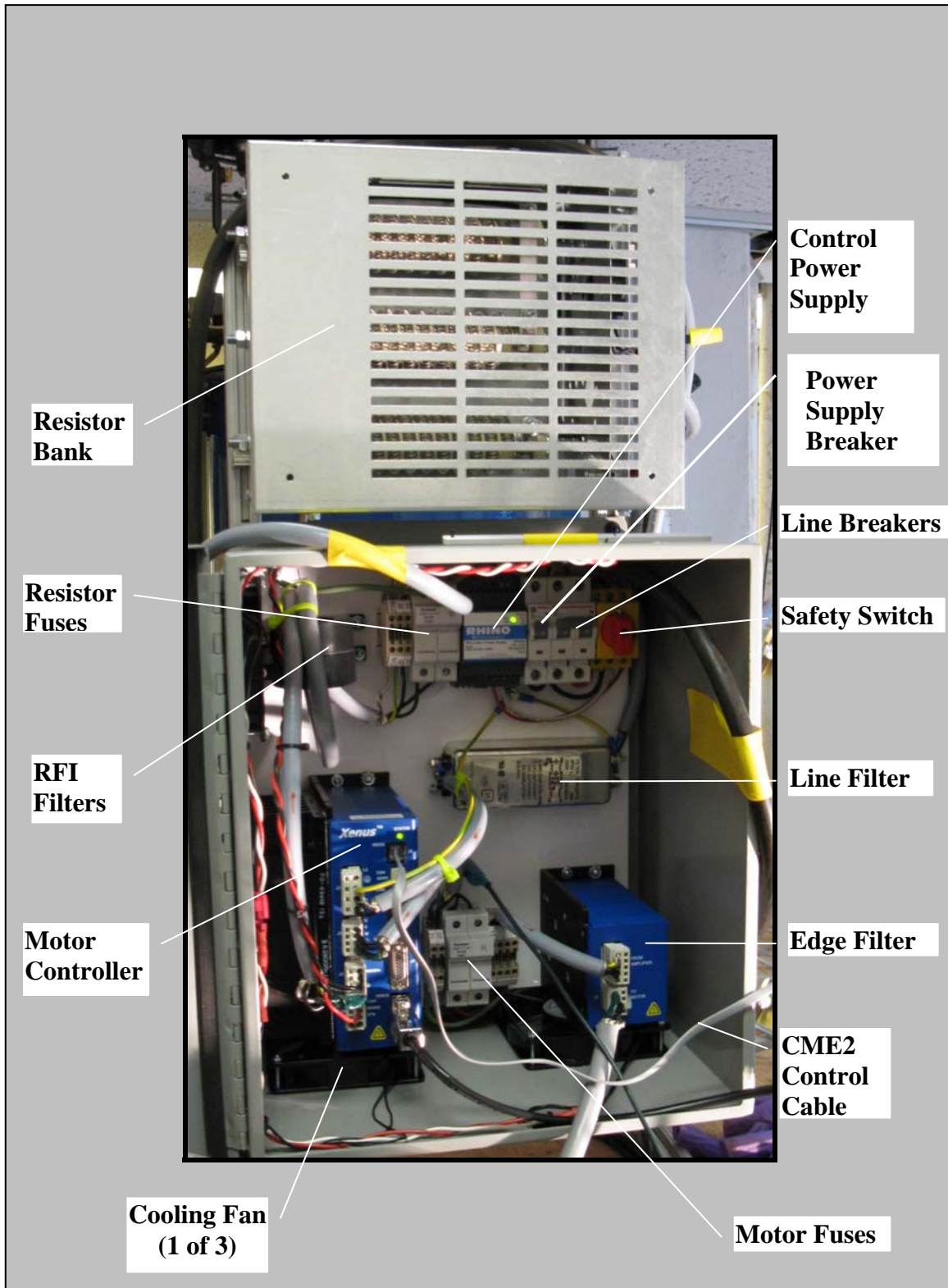


Figure 42: Electrical Components

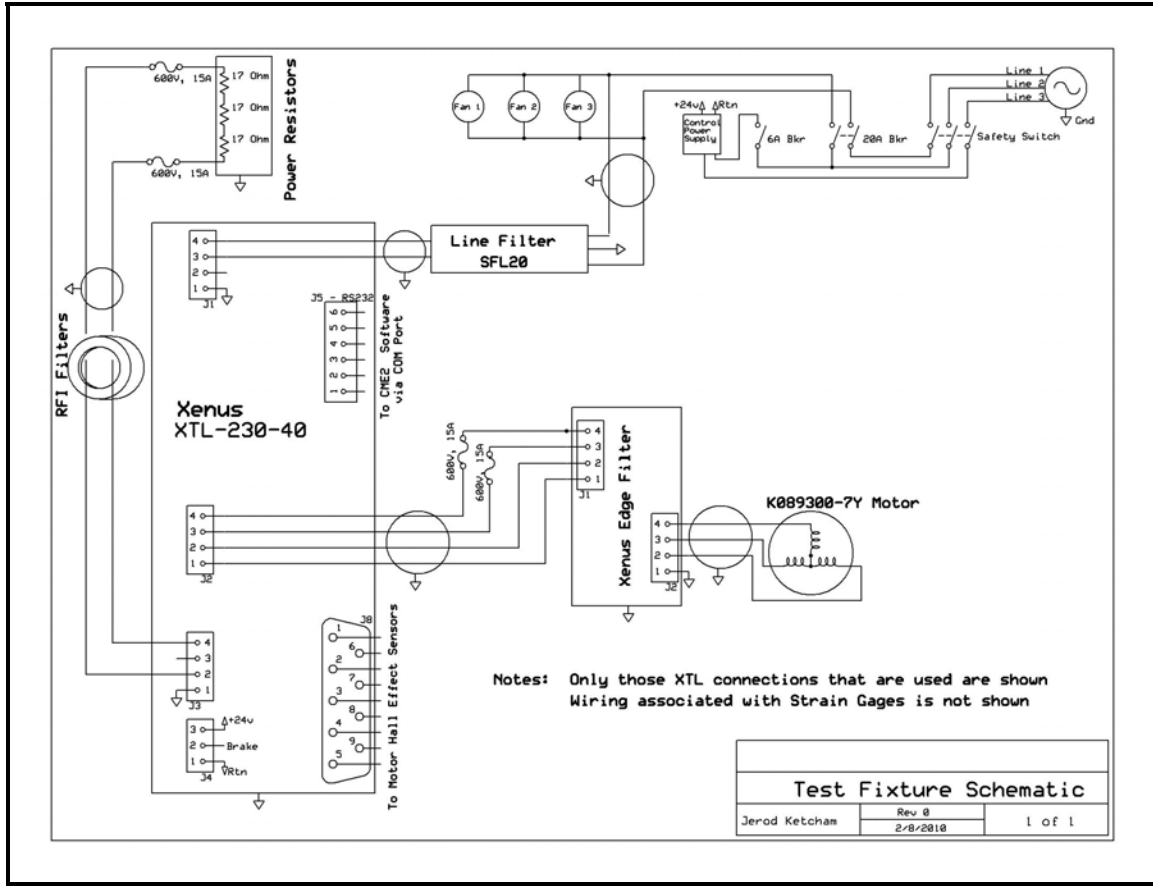


Figure 43: Schematic of Enclosure Electrical Components

A schematic of the components that are located inside the enclosure is shown in Figure 43. Note that all connection points to the motor drive are not shown, only those connections that are used are shown. Strain gage wiring is a different circuit and is not shown in Figure 43.

CONSTRUCTION

The material used in the construction of the parts of the test fixture that could be in contact with the water is stainless steel. Depending on the part, the alloy is either a 303 or 304 stainless steel. These alloys were selected for their combination of corrosion resistance and machinability. Their corrosion resistance will be sufficient for use in a fresh water environment, however prolonged use in chlorinated water and use in saltwater should be avoided to prevent corrosion. All components in the test fixture are non-magnetic with the exception of the drive shaft and propeller shaft which became slightly magnetic as a result of the machining process. A photograph of the completed test fixture in operation during turbine testing are shown in Figure 44.

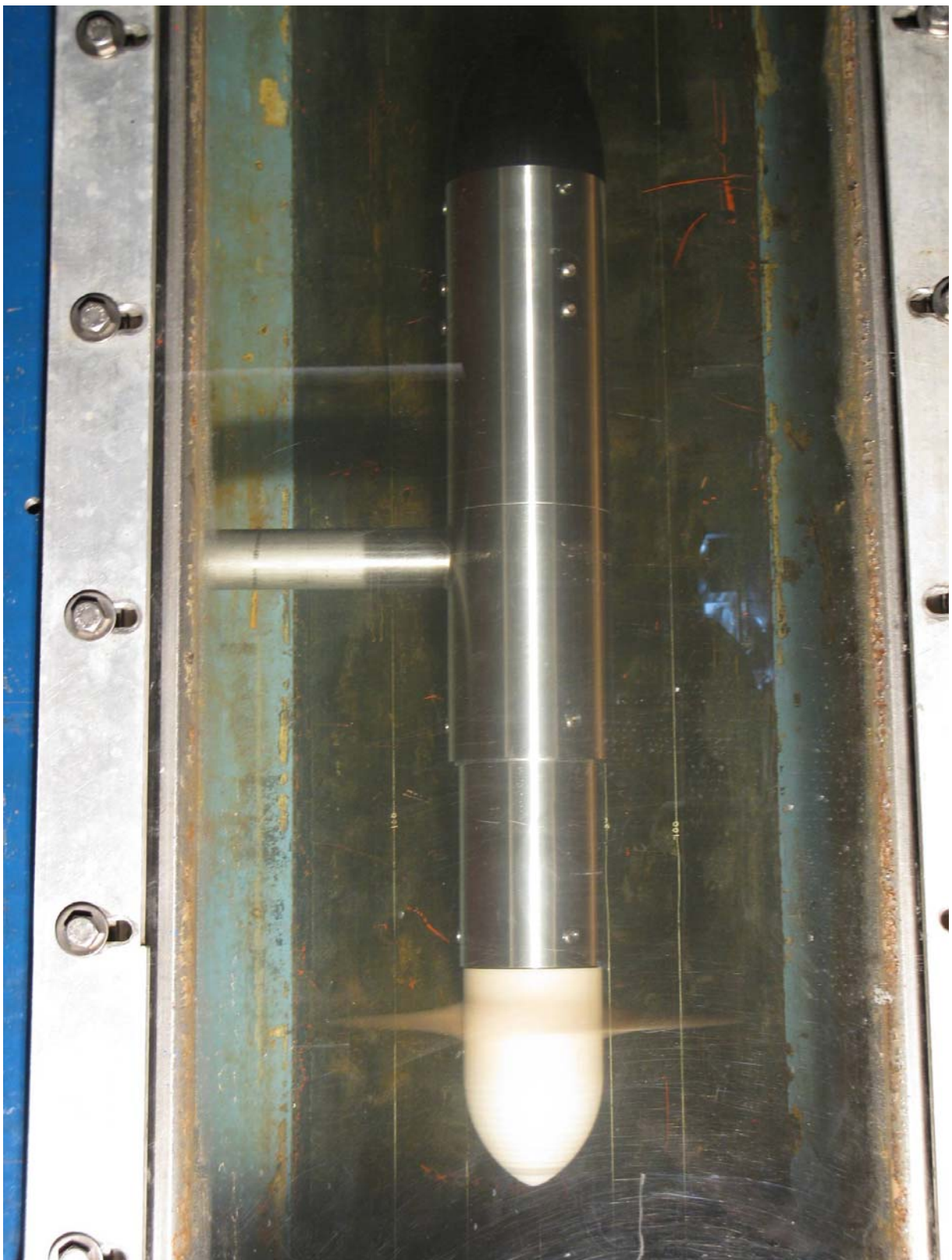


Figure 44: Completed Test Fixture in Operation

Chapter 8 – Conclusions and Recommendations

Conclusions

In this thesis a propeller and turbine test fixture was designed, constructed and used to test a model scale axial flow hydrokinetic turbine. Additionally, computer codes were written to implement ABS Steel Vessel Rules for propellers, calculate blade stresses and perform an initial estimate of fatigue life for a propeller blade.

Results from turbine tests show that the test fixture produces data signals with very little interference. Test results also show that the OpenProp performance predictions for the turbine design point are very good but that additional work is necessary to improve the off-design performance predictions of OpenProp.

Plots of blade stress distributions obtained using the methods described above show good agreement with those of Carlton (2007) which were obtained using more sophisticated techniques. The method used to calculate blade stress was easily extended to perform a fatigue analysis to obtain an initial estimate of a propeller blade turning in a wake.

Recommendations for Further Work

As currently constructed, the electrical enclosure for the test fixture must be operated with the door open in order to pass the power and control cables. The cables should be connectorized and outlets attached to the enclosure so that the door can be shut during testing.

The test procedure for collecting performance data is a manual process using LabView®. LabView® has the capability to automate the process in a way that would make data collection much quicker and the results more reliable. The test fixture could be controlled using the CME2, ASCII command interface that is part of the motor controller.

Unsteady tests would require a high resolution shaft position encoder. The test fixture was designed to allow for the inclusion of a Renishaw™ encoder. Two additional pieces would have to be manufactured; drawings for these parts are included in the appendix.

In order to implement a method to calculate blade stress, several simplifications were made. More precise results could be obtained by implementing an FEA approach. OpenProp already calculates all of the necessary geometry and load data that could be used in an FEA code. This method should be implemented with default settings that are tailored to the geometry of propeller and turbine blades such that a person with little FEA experience could obtain valid results.

The fatigue results presented assumed a quasi-steady blade response as the blade passed from one wake sector to another. More sophisticated techniques exist to estimate blade load as it passes into regions of various flow speed in the wake. These techniques should be implemented in order to refine the fatigue analysis predictions.

References

1. Benyon J.H. et al, *Engineering Against Fatigue*, A.A. Belkama, 1999
2. Carlton J.S., *Marine Propellers and Propulsion Second Edition*, Elsevier, 2007
3. Chung, H., *An Enhanced Propeller Design Program Based on Propeller Vortex Lattice Lifting Line Theory*, Master's Thesis MIT, 2007
4. Conway Joseph B., Sjødahl Lars H., *Analysis and Representation of Fatigue Data*, ASM International, 1991
5. D'Epagnier, K., *A Computational Tool for the Rapid Design and Prototyping of Propellers for Underwater Vehicles*, Master's Thesis MIT, 2007
6. Epps B., *An Impulse Framework for Hydrodynamic Force Analysis: Fish Propulsion, Water Entry of Spheres, and Marine Propellers*, PhD Thesis MIT, 2010a
7. Epps, B. P. (SM); Stanway, M. J. (SM); and Kimball, R.W. (AM)., *OpenProp: An Open-source Design Tool for Propellers and Turbines*, SNAME Propellers and Shafting, 2009
8. Epps, B., *OpenProp v2.3 Theory Document*, available at <http://openprop.mit.edu>, 2010b
9. Felli M., Felice F.D., *Propeller Wake Analysis in Nonuniform Inflow by LDV Sampling Techniques*, Journal of Marine Science and Technology, 2005
10. Gooding, T., *Principles in Naval Ship Design*, MIT course 2.703 notes, 2009
11. Kerwin J.E. and Hadler J.B., *Principles of Naval Architecture, Propulsion*, SNAME, to appear 2010
12. Kerwin, J.E., *Hydrofoils and Propellers*. MIT course 2.23 notes, 2007.
13. Kimball, R.W.; Epps, B.P.; and M.J. Stanway, *OpenProp MATLAB code*. Open-source at <http://openprop.mit.edu>
14. Koronowicz, T., Chaja, P., Szantyr J., *A Computer System for the Complete Design of Ship Propellers*, Archives of Civil and Mechanical Engineering, 2008
15. Laskos, D., *Design and Cavitation Performance of Contra-Rotating Propellers*, Master's Engineer's Thesis MIT, 2010
16. Peterson, C., *Minimum Pressure Envelope Cavitation Analysis Using Two-Dimensional Panel Method*, Master's Thesis MIT, 2008
17. Pook, L., *Metal Fatigue, What It Is, Why It Matters*, Springer 2007
18. *Rules for Building and Classing Steel Vessels 2007, Part 4 Chapter 3 Section 3*, American Bureau of Shipping, 2006
19. Shigley Joseph E., Mischke Charles R., *Mechanical Engineering Design Fifth Edition*, McGraw-Hill Inc, 1989
20. Sines, G., Waisman, J.L., *Metal Fatigue*, McGraw-Hill, 1959
21. Stubblefield, J.M., *Numerically-Based Ducted Propeller Design Using Vortex Lattice Lifting Line Theory*, Master's Thesis MIT, 2008

Appendix A – Codes

Moment of Inertia Calculation

function [Mp1, Ixc, Iyc, Ixyc, A, Xbar, Ybar, xl, yl, xu, yu] = MomentofInertia(xl,xu,yl,yu)

[Mp1,Np] = size(xu);

% Calculation of Section Area and Centroid

for m=1:Mp1

 yshift = abs(min(yl(m,:))); %Distance to shift all y points so that all are positive

 yu(m,:) = yu(m,:) + yshift; %Shift of upper surface y points

 yl(m,:) = yl(m,:) + yshift; %Shift of lower surface y points

 xshift = abs(min(min(xu(m,:)),min(xl(m,:)))); %Distance to shift all x points so that all are positive

 xu(m,:) = xu(m,:) + xshift; %Shift of upper surface x points

 xl(m,:) = xl(m,:) + xshift; %Shift of lower surface x points

end

dxu = abs(diff(xu,1,2));

dxl = abs(diff(xl,1,2));

dyu = diff(yu,1,2);

dyl = diff(yl,1,2);

Ybar = zeros(1,Mp1);

Xbar = Ybar;

Ixc = Ybar;

Iyc = Ybar;

A = Ybar;

Ixyc = Ybar;

for m=1:Mp1

 hru = zeros(1,(Np-1));

 hrl = hru;

 htu = zeros(1,(Np-1));

 htl = htu;

 xctu = zeros(1,(Np-1));

 xctl = xctu;

for n=1:(Np-1)

 hru(n)=min(yu(m,n),yu(m,n+1)); %Height of upper surface elemental rectangle

 htu(n)=max(yu(m,n),yu(m,n+1)); %Height of upper surface elemental trapezoid

 hrl(n)=min(yl(m,n),yl(m,n+1)); %Height of lower surface elemental rectangle

 htl(n)=max(yl(m,n),yl(m,n+1)); %Height of lower surface elemental trapezoid


```

if dyu(m,n)<0
    xctu(n) = xu(m,n) + 2*dxu(m,n)/3;    %Distance from y-axis to upper surface
                                          %elemental triangle centroid
else
    xctu(n) = xu(m,n) + dxu(m,n)/3;      %Note: Value depends on whether left or right
                                          %side of triangle is higher
end

if dyl(m,n)>0
    xctl(n) = xl(m,n) + 2*dxl(m,n)/3;    %Distance from y-axis to lower surface
                                          %elemental triangle centroid
else
    xctl(n) = xl(m,n) + dxl(m,n)/3;      %Note: Value depends on whether left or right
                                          %side of triangle is higher
end

end

xcru = xu(m,1:(Np-1))+dxu(m,:)/2;      %Distance from y-axis to upper surface
                                          %elemental rectangle
xcrl = xl(m,1:(Np-1))+dxl(m,:)/2;      %Distance from y-axis to lower surface
                                          %elemental rectangle

aru = dxu(m,:).*hru;                    %Elemental upper surface rectangle area
atu = dxu(m,:).*(htu-hru)/2;            %Elemental upper surface triangle area
arl = dxl(m,:).*hrl;                    %Elemental lower surface rectangle area
atl = dxl(m,:).*(htl-hrl)/2;            %Elemental lower surface triangle area

ycru = hru/2;                            %Distance from x-axis to upper surface elemental rectangle
                                          %centroid
ycrl = hrl/2;                            %Distance from x-axis to lower surface elemental rectangle
                                          %centroid
yctu = hru+(htu-hru)/3;                 %Distance from x-axis to upper surface elemental triangle
                                          %centroid
yctl = hrl+(htl-hrl)/3;                 %Distance from x-axis to lower surface elemental triangle
                                          %centroid

Mxsu = sum(ycru.*aru + yctu.*atu);      % 1st moment of upper surface about x axis
Mxsl = sum(ycrl.*arl + yctl.*atl);      % 1st moment of lower surface about x axis
Mxs = Mxsu - Mxsl;

Mysu = sum(xcru.*aru + xctu.*atu);      % 1st moment of upper surface about y axis
Mysl = sum(xcrl.*arl + xctl.*atl);      % 1st moment of lower surface about y axis
Mys = Mysu - Mysl;

```

```

Au = sum(aru + atu);      % Area of upper surface (x axis to upper surface)
Al = sum(arl + atl);      % Area of lower surface (x axis to lower surface)
A(m) = Au - Al;

```

```

Ybar(m) = Mxs/A(m);      % Distance to centroid from x-axis
Xbar(m) = Mys/A(m);      % Distance to centroid from y-axis

```

```

% Uncomment lines below to see a section graph with centroidal axes
% figure(m)
% plot(xu(m,:),yu(m,:),xl(m,:),yl(m,:), 'b')
% line([min(xu(m,:)),max(xu(m,:))],[Ybar(m),Ybar(m)], 'Color','r','LineWidth',2,'LineStyle','-')
% line([Xbar(m),Xbar(m)],[min(yl(m,:)),max(yu(m,:))], 'Color','r','LineWidth',2,'LineStyle','-')
% axis equal
% grid on

```

% Calculation of Section Moment of Inertia

```

ixru = dxu(m,:).*hru.^3/3;
ixyru = aru.*ycru.*xcru;
ixtu = dxu(m,:).*(htu-hru).^3/36 + atu.*yctu.^2;
ixytu = atu.*yctu.*xctu;

ixrl = dxl(m,:).*hrl.^3/3;
ixyrl = arl.*ycrl.*xcrl;
ixtl = dxl(m,:).*(htl-hrl).^3/36 + atl.*yctl.^2;
ixytl = atl.*yctl.*xctl;

iyru = hru.*dxu(m,:).^3/12 + aru.*xcru.^2;
iytu = (htu - hru). *dxu(m,:).^3/36 + atu.*xctu.^2;
iyrl = hrl.*dxl(m,:).^3/12 + arl.*xcrl.^2;
iytl = (htl - hrl). *dxl(m,:).^3/36 + atl.*xctl.^2;

Ix = sum(ixru + ixtu) - sum(ixrl + ixtl);
Iy = sum(iyru + iytu) - sum(iyrl + iytl);
Ixy = sum(ixyru + ixytu) - sum(ixyrl + ixytl);

Ixc(m) = Ix - A(m)*Ybar(m)^2;
Iyc(m) = Iy - A(m)*Xbar(m)^2;
Ixy(m) = Ixy - A(m)*Xbar(m)*Ybar(m);
end

```

end

Centrifugal Force Calculation

function [omega, Fc, Rdif] = CentrifugalForce(Mp1, N, R, A, RC, DR)

%Initialization of Variables

Fc = zeros(1,Mp1-1);

Rdif = zeros(Mp1-1,Mp1-1);

R1 = Rdif;

omega = 2*pi*N/60; % [rev/s] Rotation rate

gamma = 1024.16; % [kg/m^3] Density of ABS plastic

% 8200; % [kg/m^3] Approximate density of Ni-Al-Bronze

M = (omega)^2 * gamma * R^2; %Multiplier used below

%Calculation of centrifugal force on each section

for m = 1:(Mp1-1)

 Fc(m) = M * sum(A(m:end-1).*RC(m:end).*DR(m:end)); % [N]

 R1(m,:) = RC - RC(m);

 I = find(R1(m,:) > 0);

 Rdif(m,I) = R1(m,I);

end

end

Stress Calculation

function [s] = Stress(CD, CL, BetaIC, Mp1, Np, xu, yu, xl, yl, rho, Vs, R, VSTAR, CoD, Rdif, DR, theta, Xbar, Ybar, Ixc, Iyc, Ixyc, Fc, A)

```
% Uncomment lines below to use Kerwin and Hadler method exactly
% eps = CD./CL;
% S = sin(BetaIC-eps);
% C = cos(BetaIC-eps);
```

```
S = sin(BetaIC); %Factors for use below
C = cos(BetaIC); %Factors for use below
```

%Initialization of Variables

```
INTQ = zeros(Mp1-1,Mp1-1);
INTT = INTQ;
MQ = zeros(1,Mp1-1);
MT = MQ;
Mxo = MQ;
Myo = MQ;
s = zeros(Mp1-1,2*Np);
```

%Concatenation of upper and lower section curves into a single curve

```
Xu = xu(:,:);
Yu = yu(:,:);
xs = cat(2,Xu,fliplr(xl));
ys = cat(2,Yu,fliplr(yl));
```

%Uncomment lines below to see a plot of root blade section

```
% axes('fontweight','bold')
% hold on
% plot(xs(1,:),ys(1,:),'-ks','LineWidth',2)
% axis equal
% title('Root Section Plot','Fontweight','bold','FontSize',14)
% xlabel('X (m)','Fontweight','bold','FontSize',12)
% ylabel('Y (m)','Fontweight','bold','FontSize',12)
```

```
M = rho*Vs^2*R^3; %Multiplier used below
for m=1:(Mp1-1)
```

%Uncomment two lines below to use Kerwin and Hadler method exactly

```
%INTQ(m,:) = M* (VSTAR.^2 .* CL .* CoD .* Rdif(m,:) .*S .*DR);
%INTT(m,:) = M* (VSTAR.^2 .* CL .* CoD .* Rdif(m,:) .*C .*DR);
INTQ(m,:) = M* (VSTAR.^2 .* (CL .*S + CD.*C).* CoD .* Rdif(m,:).*DR);
INTT(m,:) = M* (VSTAR.^2 .* (CL .*C - CD.*S).* CoD .* Rdif(m,:).*DR);
```

```
MQ(m) = sum(INTQ(m,:));
```

```

MT(m) = sum(INTT(m,:));
Mxo(m) = MT(m)*cos(theta(m)) + MQ(m)*sin(theta(m));
Myo(m) = MT(m)*sin(theta(m)) - MQ(m)*cos(theta(m));

xsdiff(m,:) = xs(m,:) - Xbar(m);

ysdiff(m,:) = ys(m,:) - Ybar(m);

s(m,:) = ((-Mxo(m)*Iyc(m) + Myo(m)*Ixyc(m))*ysdiff(m,:) - (-Mxo(m)*Ixyc(m) +
Myo(m)*Ixc(m))*xsdiff(m,:)) / (Ixc(m)*Iyc(m) - Ixyc(m)^2) + Fc(m)/A(m);

%Uncomment line below to use a more exact equation for stress which
%takes into account the product of inertia
s(m,:) = ((-Mxo(m)*Iyc(m) + Myo(m)*Ixyc(m))*ys(m,:) - (-Mxo(m)*Ixyc(m) +
Myo(m)*Ixc(m))*xs(m,:)) / (Ixc(m)*Iyc(m) - Ixyc(m)^2) + Fc(m)/A(m);

%Uncomment lines below for plots of stress on each blade section
% figure(m)
% % plot3(xs(m,:),ys(m,:),s(m,:), 'rs')
% % grid on
% % xlim([min(xs(m,:)),max(xs(m,:))])
% % ylim(xlim)
% patch(xs(m,:),ys(m,:),s(m,:))
% colormap(jet)
% colorbar
% grid on
% axis equal
end
clear CL

end

```

Blade Stress Plots

```
function [ ] = Plot_Blade_Contours(X3D,Y3D,Z3D,s,plottitle)
```

```
    [rows,cols]=size(X3D); %X3D is from the geometry.m module
```

```
    Mp = rows-1;          %Number of blade sections
```

%Concatenate matrices to create vertex matrix for patch function. This is necessary because the patch function expects a matrix whose rows are the vertices in x,y,z coordinates.

```
    colx = X3D(1,:);
```

```
    coly = Y3D(1,:);
```

```
    colz = Z3D(1,:);
```

```
    colS = s(1,:);
```

```
    for n=2:rows-1
```

```
        colx = vertcat(colx,X3D(n,:));
```

```
        coly = vertcat(coly,Y3D(n,:));
```

```
        colz = vertcat(colz,Z3D(n,:));
```

```
        colS = vertcat(colS,s(n,:));
```

```
    end
```

%Create face matrix for patch function- this tells the patch function how to connect the vertices to create a face. This code uses a square/rectangular face.

```
    F(:,1) = 1:cols*(Mp-1);
```

```
    F(:,2) = F(:,1) + 1;
```

```
    F(:,3) = F(:,1) + (cols+1);
```

```
    F(:,4) = F(:,1) + cols;
```

%Create special rows - the pattern of face vertices "wraps," these lines make the pattern wrap properly

```
    m = 0:cols:cols*(Mp-1);
```

```
    for n = 1:length(m)
```

```
        if m(n)>1
```

```
            F(m(n),1:4) = [F(m(n),1), F(m(n),1)-(cols-1), F(m(n),1)+1, F(m(n),1)+4];
```

```
        end
```

```
    end
```

%Remove extra face matrix rows - This removes "extra" rows from wrapping scheme above

```
    Fa = F(1:cols-1,:);
```

```
    for o=1:Mp-2
```

```
        Fa = vertcat(Fa, F(o*cols+1:(o+1)*cols-1,:));
```

```
    end
```

% Create Vertices matrix

```
    a = [colx,coly,colz];
```

```

% Create Figure
figure1 = figure('Color',[1 1 1]);

% Create axes
axes('Visible','off','Parent',figure1,'CLim',[-4e+006 4e+006]);
view([-83 2]);
colorbar('FontWeight','bold')
title(plottitle,'Visible','On','FontSize',14,'FontWeight','bold')

% Call patch function
patch('Vertices',a,'Faces',Fa,'FaceVertexCData',colS,'FaceColor','interp','FaceLighting','gouraud')

% Uncomment line below and adjust numbers to set colorbar scale
% axis('CLim',[-2e+008 2e+008])

end

```

Usage Script

Load Variables

```
D = pt.geometry.D;           %Propeller Diameter
N = pt.geometry.N;           %Propeller rotation rate [RPM]
RC = pt.design.RC;           %Control Point (Section) Radius []
DR = pt.design.DR;           %Control Point Radius Difference []
VSTAR = pt.design.VSTAR;     %Total inflow velocity []
TANBC = pt.design.TANBC;     %Tangent of inflow angle for each Control Point
BetaIC = pt.design.BetaIC;   %Ideal inflow angle for each Control Point
CL = pt.design.CL;           %On-design Lift Coefficient for each Control Point
CD = pt.design.CD;           %On-design Drag Coefficient for each Control Point
CoD = pt.design.CoD;         %Chord Length/Diameter for each Control Point
xu = pt.geometry.xu;         %Upper section x points - Leading edge to trailing edge
yu = pt.geometry.yu;         %Upper section y points - Leading edge to trailing edge
xl = pt.geometry.xl;         %Lower section x points - Leading edge to trailing edge
yl = pt.geometry.yl;         %Lower section y points - Leading edge to trailing edge
X3D = pt.geometry.X3D;       %x points which create blade surface
Y3D = pt.geometry.Y3D;       %y points which create blade surface
Z3D = pt.geometry.Z3D;       %z points which create blade surface
UASTAR = pt.design.UASTAR;   %Axial blade influence velocity []
UTSTAR = pt.design.UTSTAR;   %Tangential blade influence velocity []
Vs = pt.input.Vs;            %Design ship speed [m/s]
Js = pt.input.Js;            %Design Advance Coefficient
alpha = pt.design.alpha;     %[deg] Section angle of attack

theta = pi/180 * pt.geometry.theta; %Pitch angle in radians
```

Section Centroid and Moments of Inertia Calculation

```
[Mp1, Ixc, Iyc, Ixyc, A, Xbar, Ybar, xl, yl, xu, yu] = MomentofInertia(xl,xu,yl,yu);
```

Centrifugal Force

```
[omega, Fc, Rdif] = CentrifugalForce(Mp1, N, R, A, RC, DR);
```

Stress Calculation

```
[s] = Stress(CD, CL, BetaIC, Mp1, Np, xu, yu, xl, yl, rho, Vs, R, VSTAR, CoD, Rdif, DR, theta, Xbar, Ybar, Ixc, Iyc, Ixyc, Fc, A);
```

Make Stress Plot

```
Plot_Blade_Contours(X3D, Y3D, Z3D, s, 'Suction Side')
```


Blade Thickness ABS

function [stress] = StructureABS(pt)

Unpack Variables

```
Date_string = pt.date;
filename     = pt.input.filename;
Make2Dplot_flag = pt.input.Make2Dplot_flag;
Make3Dplot_flag = pt.input.Make3Dplot_flag;
Make_Rhino_flag = pt.input.Make_Rhino_flag;
Meanline     = pt.input.Meanline;
Thickness    = pt.input.Thickness;
XR           = pt.input.XR;
f0oc0        = pt.input.f0oc0;
t0oc0        = pt.input.t0oc0;
XCoD         = pt.input.XCoD;
skew0        = pt.input.skew0; % [deg]
rake0        = pt.input.rake0;
alphaI       = pt.input.alphaI; % [deg]
CLI          = pt.input.CLI;
Z            = pt.input.Z;
Js           = pt.input.Js;
Vs           = pt.input.Vs; % [m/s]
R            = pt.input.R; % [m]
Rhub         = pt.input.Rhub; % [m] hub radius
Np           = pt.input.Np;
H_flag       = pt.input.H_flag;
D_flag       = pt.input.D_flag;
RC           = pt.design.RC;
RV           = pt.design.RV;
CL           = pt.design.CL;
Beta_c       = atand(pt.design.TANBC); % [deg]
BetaI_c      = pt.design.BetaIC*180/pi; % [deg]
CoD          = pt.design.CoD;
TAU          = pt.design.TAU;
TANBIV       = pt.design.TANBIV;
D            = 2*R; % [m]
Dhub         = 2*Rhub; % [m]
Rhub_oR      = Rhub/R;
N            = 60*Vs/(Js*D); % [RPM]
```

Interpolate input geometry at selected radial sections

% Set vortex points 1 and Mp+1 to the hub and tip: RG = [0.9*Rhub_oR,RV(2:end-1),1];

Interpolate Input Geometry at Sections with Cosine Spacing Along the Span

RG = [0.25,0.70]; % Required sections for ABS analysis

Mp = length(RG)-1;

```
f0oc = pchip(XR,f0oc0,RG).*pchip(RC,CL,RG)/CLI; % [ ], camber ratio scaled with lift
coefficient
```

```
skew = pchip(XR,skew0,RG); % [deg], angular translation along mid-chord helix
rake = pchip(XR,rake0,RG)*D; % [m], translation along propeller axis (3D X-axis)
```

```
% t0oc = pchip(XR,t0oc0,RG); % [ ], thickness ratio
% CoD = interp1(RC,CoD,RG,'pchip','extrap');
```

```
CoD = pchip(XR,XCoD,RG);
```

```
% Thesis4: thickness profile
```

```
TTRF = 0.5; % Tip Thickness Reduction Factor == modified thickness at tip / baseline
thickness at tip
```

```
XRmax = 0.8; % maximum XR for which thickness reduction is less than 1%
```

```
HTTR = 3.5; % Hub-Tip Thickness Ratio = t0(hub) / t0(tip)
```

```
t0tip = 0.00254; % [m] == 0.254 mm = 0.1 inch, max thickness at tip section
```

```
t0oc = t0tip*(HTTR - (HTTR-1).*(RG-Dhub/D)/(1-Dhub/D))./(CoD*D) .* (1-(1-TTRF)*exp(-
4.6*(1-RG)/(1-XRmax)));
```

Find Basic Geometry Parameters Chord, Radius, Pitch, etc.

```
theta = interp1(RC,BetaI_c+alphaI.*CL/CLI,RG,'pchip','extrap'); % Nose-tail pitch angle,
[deg]
```

```
PoD = tand(theta).*pi.*RG; % Pitch / propeller diameter, [ ]
```

```
c = CoD.*D; % section chord at the sections [m]
```

```
r = RG.*R; % radius of the sections [m]
```

```
theta_Z = 0:360/Z:360; % angle between blades [deg]
```

Lay Out the 2D Coordinate System

```
xN [ ], x/c coordinate in 2D NACA foil tables
```

```
At the Leading Edge: xN = 0, x1 = c/2, x0 = 0
```

```
At the Trailing Edge: xN = 1, x1 = -c/2, x0 = 1
```

```
x0 [ ], x/c distance along mid-chord line to interpolate NACA foil table data.
```

```
x1 [m], x distance along mid-chord line to evaluate elliptical or parabolic formulae. By
definition, x1 == c/2 - c*x0.
```

```
x2D [m], x position in 2D space on upper (x2D_u) and lower (x2D_l) foil surfaces
```

```
y2D [m], y position in 2D space on upper (x2D_u) and lower (x2D_l) foil surfaces
```

```
x2Dr [m], x position in 2D space after rotation for pitch angle
```

```
y2Dr [m], y position in 2D space after rotation for pitch angle
```

```
xN = [0 .5 .75 1.25 2.5 5 7.5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100]./100;
```

```
x0 = zeros(1,Np);
```

```
x1 = zeros(Mp+1,Np);
```

% % Even spacing along the chord

```
% % for i = 1:Mp          % for each radial section along the span
% for i = 1:Mp+1         % for each radial section along the span
%   for j = 1:Np          % for each point      along the chord
%       x0(1,j) = (j-1)/(Np-1); % [0 : 1]
%       x1(i,j) = c(i)/2 - c(i)*(j-1)/(Np-1); % [c/2 : -c/2]
%   end
% end
```

% Cosine spacing along the chord

```
for i = 1:Mp+1          % for each radial section along the span
    for j = 1:Np          % for each point      along the chord
        x1(i,j) = c(i)/2 - 0.5*c(i)*(1-cos(pi*(j-1)/(Np-1))); % [c/2 : -c/2]
    end
end
x0 = 0.5-x1(1,:)/c(1);
```

Find Meanline and Thickness Profiles (at x1 positions)

```
foc = camber / chord ratio (NACA data at xN positions)
dfdxN = slope of camber line (NACA data at xN positions)
fscale = scale to set max camber ratio to f0oc for each section
tscale = scale to set max thickness ratio to t0oc for each section
f = camber at x1 positions
dfdx = slope of camber line at x1 positions
t = thickness at x1 positions

t = zeros(Mp+1,Np);
f = zeros(Mp+1,Np);
dfdx = zeros(Mp+1,Np);
```

```
if Meanline==0 | strcmp(Meanline,'NACA a=0.8 (modified)') % Use NACA a=0.8 (modified)
    meanline
```

```
    foc = [0 0.281 0.396 0.603 1.055 1.803 2.432 2.981 3.903 4.651 5.257 5.742 6.120 6.394
6.571 6.651 6.631 6.508 6.274 5.913 5.401 4.673 3.607 2.452 1.226 0 ]./100;
```

```
    dfdxN = [0 0.47539 0.44004 0.39531 0.33404 0.27149 0.23378 0.20618 0.16546 0.13452
0.10873 0.08595 0.06498 0.04507 0.02559 0.00607 -0.01404 -0.03537 -0.05887 -0.08610
-0.12058 -0.18034 -0.23430 -0.24521 -0.24521 -0.24521];
```

```
    fscale = f0oc / max(foc);
    dfdxLE = 0.47539*fscale; % slope at leading edge
```

```
for i = 1:Mp+1          % for each radial section along the span
    for j = 1:Np
        f(i,:) = pchip(xN,foc .*fscale(i).*c(i),x0);
```

```

        dfdx(i,:) = pchip(xN,dfdxN.*fscale(i) ,x0);
    end
end

elseif Meanline==1 | strcmp(Meanline,'NACA a=0.8') % Use NACA a=0.8 meanline
    foc = [0 .287 .404 .616 1.077 1.841 2.483 3.043 3.985 4.748 5.367 5.863 6.248 6.528 6.709
6.790 6.770 6.644 6.405 6.037 5.514 4.771 3.683 2.435 1.163 0]./100;

    dfdxN = [0 .48535 .44925 .40359 .34104 .27718 .23868 .21050
16892 .13734 .11101 .08775 .06634 .04601 .02613 00620 -.01433 -.03611 -.06010 -.08790 -
.12311 -.18412 -.23921 -.25583 -.24904 -.20385];

    fscale = f0oc / max(foc);
    dfdxLE = 0.48535*fscale; % slope at leading edge

    for i = 1:Mp+1 % for each radial section along the span
        for j = 1:Np
            f(i,:) = pchip(xN,foc .*fscale(i).*c(i),x0);
            dfdx(i,:) = pchip(xN,dfdxN.*fscale(i) ,x0);
        end
    end

% elseif Meanline==2 | strcmp(Meanline,'parabolic') % Use parabolic meanline
%
% % for i = 1:Mp % for each radial section along the span
% for i = 1:Mp+1 % for each radial section along the span
%     for j = 1:Np
%         f(i,j) = f0oc(i)*c(i)*(1-(2*x1(i,j)/c(i))^2);
%         dfdx(i,j) = -8*f0oc(i)*x1(i,j)/c(i);
%     end
% end

if Thickness==1 | strcmp(Thickness,'NACA 65A010') % Use NACA 65A010 thickness form
    toc_65 = [0 .765 .928 1.183 1.623 2.182 2.65 3.04 3.658 4.127 4.483 4.742 4.912 4.995 4.983
4.863 4.632 4.304 3.899 3.432 2.912 2.352 1.771 1.188 .604 .021]./100;

    tscale = t0oc / max(toc_65);

    rLE = 0.00639*c.*tscale; % leading edge radius

    for i = 1:Mp+1 % for each radial section along the span
        for j = 1:Np
            t(i,:) = pchip(xN,toc_65.*tscale(i).*c(i),x0);
        end
    end
end

```

```

elseif Thickness==2 | strcmp(Thickness,'elliptical') % Use elliptical thickness form
    for i = 1:Mp+1          % for each radial section along the span
        for j = 1:Np
            t(i,j) = t0oc(i)*c(i)*real(sqrt(1-(2*x1(i,j)/c(i))^2));
        end
    end

    rLE = 0; % leading edge radius

elseif Thickness==3 | strcmp(Thickness,'parabolic') % Use parabolic thickness form
    for i = 1:Mp+1          % for each radial section along the span
        for j = 1:Np
            t(i,j) = t0oc(i)*c(i)*(1-(2*x1(i,j)/c(i))^2);
        end
    end

    rLE = 0; % leading edge radius

elseif Thickness==4 | strcmp(Thickness,'NACA 65A010 (modified)') % Use modified NACA
65A010 thickness form
    xx65mod = [0  0.0050000000000000  0.0075000000000000  0.0125000000000000
0.0250000000000000  0.0500000000000000  0.0750000000000000  0.1000000000000000
0.1500000000000000  0.2000000000000000  0.2500000000000000  0.3000000000000000
0.3500000000000000  0.4000000000000000  0.471204188481675  0.523560209424084
0.575916230366492  0.628272251308901  0.680628272251309  0.732984293193717
0.785340314136126  0.837696335078534  0.890052356020942  0.942408376963351
0.968586387434555  0.981675392670157  0.989528795811518  0.994764397905759
0.997382198952880  1.0000000000000000];

    tt65mod = [0  0.0076500000000000  0.0092800000000000  0.0118300000000000
0.0162300000000000  0.0218200000000000  0.0265000000000000  0.0304000000000000
0.0365800000000000  0.0412700000000000  0.0448300000000000  0.0474200000000000
0.0491200000000000  0.0499500000000000  0.0498300000000000  0.0486300000000000
0.0463200000000000  0.0430400000000000  0.0389900000000000  0.0343200000000000
0.0291200000000000  0.0235200000000000  0.0177100000000000  0.0118800000000000
0.0089600000000000  0.007499530848329  0.006623639691517  0.0060400000000000
0.004049015364794  0.0002100000000000];

    tscale = t0oc / max(tt65mod);

    rLE = 0.00639*c.*tscale; % leading edge radius

    for i = 1:Mp+1          % for each radial section along the span
        for j = 1:Np
            t(i,:) = pchp(xx65mod,tt65mod.*tscale(i).*c(i),x0);

```

```

    end
end

```

end

Find 2D Unrotated Section Profiles

x2D [m], x position in 2D space on upper (x2D_u) and lower (x2D_l) foil surfaces
y2D [m], y position in 2D space on upper (x2D_u) and lower (x2D_l) foil surfaces

```

x2D_u = zeros(Mp+1,Np);    x2D_l = zeros(Mp+1,Np);
y2D_u = zeros(Mp+1,Np);    y2D_l = zeros(Mp+1,Np);

for i = 1:Mp+1              % for each section along the span
    for j = 1:Np             % for each point along the chord
        x2D_u(i,j) = x1(i,j) + (t(i,j)/2)*sin(atan(dfdx(i,j))); % 2D upper surface x
        x2D_l(i,j) = x1(i,j) - (t(i,j)/2)*sin(atan(dfdx(i,j))); % 2D lower surface x
        y2D_u(i,j) = f(i,j) + (t(i,j)/2)*cos(atan(dfdx(i,j))); % 2D upper surface y
        y2D_l(i,j) = f(i,j) - (t(i,j)/2)*cos(atan(dfdx(i,j))); % 2D lower surface y
    end
end

```

% Put all the numbers in one list

```

% % j = 1      == tail
% % j = 1:Np   == suction side
% % j = Np     == nose
% % j = Np + 1 == nose
% % j = Np+ 1:2*Np == pressure side
% % j = 2*Np   == tail
% % Tail -> suction side -> nose, nose -> pressure side -> tail
x2D(:, 1:Np ) = x2D_u(:,Np:-1:1); % The first Np values are the upper surface (suction side),
x2D(:,Np+1:Np+Np) = x2D_l(:,1:Np); % and the second Np values are the lower surface
                                   (pressure side).

y2D(:, 1:Np ) = y2D_u(:,Np:-1:1);
y2D(:,Np+1:Np+Np) = y2D_l(:,1:Np);

```

Find power at design power:

```

Pdesign = pt.design.CP * (1/2) * pt.input.rho * Vs^3 * pi*R^2;

```

Find expanded area ratio

```

EAR = Z * trapz(XR*R,XCoD*D) / (pi*R^2);

```

Variable Definitions

```

% a - Expanded blade area divided by disc area
% a_s - Area of expanded cylindrical section at 0.25 radius (mm^2)
% C_n - Section modulus coefficient at the 0.25 radius. To be determined
% using the equation listed in the code below.

```

```

% C_s - Section area coefficient at 0.25 radius and is to determined by the
% equation listed in the code below.
% D - Propeller Diameter (m)
% f, w - Material constants
% H - Power at Rated Speed (kW)
% I_o - Moment of inertia of expanded cylindrical section at 0.25 radius
% about a line through the center of gravity parallel to the pitch line or
% to the nose (mm^4)
% K - Rake of propeller blade in mm (positive for aft rake and negative for
% forward rake
% K1 - Coefficient 337 for SI units
% N - Number of blades
% P25R - Pitch at 1/4 radius divided by propeller diameter,
% corresponding to the design ahead condition
% P25 - Pitch at 1/4 radius
% P70R - Pitch at 7/10 radius divided by propeller diameter,
% corresponding to the design ahead condition
% P70 - Pitch at 7/10 radius
% R - rpm at rated speed
% S - Factor
% t25 - minimum required thickness at the thickest part of the blade
% section at one quarter radius (mm)
% T - Maximum designed thickness of blade section at 0.25 radius from
% propeller drawing (mm)
% theta - Pitch angle in radians
% U_f - Maximum nominal distance from the moment of inertia axis to points
% of the face boundary (tension side) of the section (mm)
% W - Expanded width of a cylindrical section at 0.25 radius (mm)

```

Material Data

```

% matl=input('Enter a material type from the list below:\n Manganese bronze\n Nickel-
manganese bronze\n Nickel-aluminum bronze\n Manganese-nickel-aluminum bronze\n Stainless
steel\n \'n','s');
% if strcmp(matl,'Manganese bronze')
% f=2.10; w=8.3;
% elseif strcmp(matl,'Nickel-manganese bronze')
% f=2.13; w=8.0;
% elseif strcmp(matl,'Nickel-aluminum bronze')
% f=2.62; w=7.5;
% elseif strcmp(matl,'Manganese-nickel-aluminum bronze')
% f=2.37; w=7.5; % elseif strcmp(matl,'Stainless steel')
% f=2.10; w=7.75; % end

f = 2.62;
w = 7.5;

```

Data Imported from other OpenProp Modules

```
% x2D(i,j) == [m] jth point at ith blade section
% y2D(i,j) == [m] jth point at ith blade section
% i == 1 for r/R == 0.25
% i == 2 for r/R == 0.70
% Tail -> suction side -> nose, nose -> pressure side -> tail
% j = 1 == tail
% j = 1:Np == suction side
% j = Np == nose
% j = Np + 1 == nose
% j = Np+ 1:2*Np == pressure side
% j = 2*Np == tail
% Pdesign == [W] power required at the design state
% Z == number of blades
% D == [m], diameter
% N == [RPM], rotation rate
% EAR == [ ], expanded blade area / disc area
```

%Points of foil section at r=0.25R and r=0.70R

```
for n=1:2
```

```
    if n==1
```

```
        xu25 = x2D(n,1:Np)*1000;
        xl25 = x2D(n,Np+1:2*Np)*1000;
        yu25 = y2D(n,1:Np)*1000;
        yl25 = y2D(n,Np+1:2*Np)*1000;
```

```
    elseif n==2
```

```
        xu70 = x2D(n,1:Np)*1000;
        xl70 = x2D(n,Np+1:2*Np)*1000;
        yu70 = y2D(n,1:Np)*1000;
        yl70 = y2D(n,Np+1:2*Np)*1000;
```

```
    end
```

```
end
```

%Flip lower surface points for use in determining maximum blade thickness

```
xl25 = fliplr(xl25);
yl25 = fliplr(yl25);
xl70 = fliplr(xl70);
yl70 = fliplr(yl70);
```

%Power at rated speed (kW)

```
H = Pdesign/1000;
```

%RPM at rated speed

```
R = N;
```


% Number of blades

$N = Z;$

%Area ratio

$a = EAR;$

%Rake (mm)

$K = 0;$

%Pitch ratio at 1/4 and 7/10 radius

$P_{25R} = PoD(1);$

$P_{70R} = PoD(2);$

%Expanded width of 1/4 radius section (mm)

$I_{max25} = \text{find}(xu_{25} == \max(xu_{25}));$

$I_{min25} = \text{find}(xu_{25} == \min(xu_{25}));$

$W = \sqrt{(xu_{25}(I_{max25}) - xu_{25}(I_{min25}))^2 + (yu_{25}(I_{max25}) - yu_{25}(I_{min25}))^2};$

% a_s (mm²)

%Shift of section to accomodate lower section zero crossings

$y_{min25} = \min(yl_{25});$

$yu_{25} = yu_{25} - y_{min25};$

$yl_{25} = yl_{25} - y_{min25};$

$x_{min25} = \min(xl_{25});$

$xu_{25} = xu_{25} + x_{min25};$

$xl_{25} = xl_{25} + x_{min25};$

Section Area and Centroid Calculation

$A_u = \text{trapz}(xu_{25}, yu_{25});$

$A_l = \text{trapz}(xl_{25}, yl_{25});$

$a_s = (A_u - A_l);$

% Io (mm⁴)

%Centroid Calculation

$h_{1u} = \text{zeros}(1, \text{length}(yu_{25}) - 1);$

$h_{1l} = h_{1u};$

for $n = 1 : \text{length}(yu_{25}) - 1$

$h_{1u}(n) = \min(yu_{25}(n), yu_{25}(n+1));$

end

for $n = 1 : \text{length}(yl_{25}) - 1$

$h_{1l}(n) = \min(yl_{25}(n), yl_{25}(n+1));$

end

```

h2u=abs(diff(yu25));
h2l=abs(diff(yl25));
dxu25=diff(xu25);
dxl25=diff(xl25);

Mxu25=sum(dxu25/2.*(h2u.^2/3 + h1u.*h2u + h1u.^2));
Mxl25=sum(dxl25/2.*(h2l.^2/3 + h1l.*h2l + h1l.^2));

Mx=Mxu25-Mxl25;
ybar=Mx/a_s;

```

Moment of Inertia Calculation

```

Iu=sum(dxu25.*(4*h1u.^3 + h2u.^3 + 6*h1u.^2.*h2u));
Il=sum(dxl25.*(4*h1l.^3 + h2l.^3 + 6*h1l.^2.*h2l));
Ix=Iu-Il;
Io=(Ix-ybar^2*a_s);

plot(xu25,yu25,'-bs',xl25,yl25,'-rs','MarkerSize',8,'LineWidth',2)
grid on
x=[min(xu25),max(xu25)];
y=[ybar,ybar];
line(x,y,'Color','g','LineWidth',2,'LineStyle','--')

```

Maximum Distance from Neutral Axis to Tension Side (mm)

```

U_f=max(abs(ybar-yl25));

```

Maximum Designed Blade Thickness (mm)

```

no = round(Np*.25);
thick = zeros(1,Np-no);
for n=Np:-1:no;
    thick(n)=abs(yu25(n)-yl25(n));
end

```

```

T = max(thick);

```

```

[trash,Ithickmax] = max(thick);
x = [xu25(Ithickmax),xu25(Ithickmax)];
y = [yl25(Ithickmax),yu25(Ithickmax)];
line(x,y,'Color','r','LineWidth',2,'LineStyle','--')
xlim([min(xu25)-10,max(xu25)+10]);
ylim([min(yl25)-1,max(yu25)+1]);
% Iotext=num2str(Io);
text(xmin25,T/2, strcat('Io = ',num2str(fix(Io)), ' mm^2'),'fontweight','b','fontsize',12)
% annotation('textbox',[0.25,0.25,.3,.3])

```

```

title('Section at 25% R','fontsize',14,'fontweight','b')
xlabel('X2D (mm)','fontweight','b')
ylabel('Y2D (mm)','fontweight','b')
legend('suction side','pressure side', 'neutral axis', 'max thickness')

```

Determining S Factor for SI and MKS units

```

if D<=6.1
    S=1;
else
    S=sqrt((D+24)/30.1);
end

```

```

if S>1.025
    S=1.025;
end

```

Calculation of Constants and Required Blade Thickness

```

K1=337; %Constant for SI and MKS units
A=1+6/P70R+4.3*P25R;
B=(4300*w*a/N)*(R/100)^2*(D/20)^3;
C=(1+1.5*P25R)*(W*f-B);
C_n=Io/(U_f*W*T^2);
C_s=a_s/(W*T);
t25=S*(K1*sqrt(A*H/(C_n*C*R*N)))+(C_s/C_n)*B*K/4/C;

```

Output stress data structure

```

stress.Pdesign = Pdesign;
stress.Z = N;
stress.D = D;
stress.N = R;
stress.EAR = EAR;
stress.t25 = t25;
stress.T = T;

```

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B – Parts List

Line Number	Source	Part Name	Part Description	Part Number / Model Number	Qty.	Individual Cost	Total Cost
1	MSCdirect	Propeller Shaft Seal	Inside Diameter: 0.438" Outside Diameter: 0.875" Seal Type: SM Material: NITRILE Thickness: 0.250"	36676138	1	\$3.03	\$3.03
2	MSCdirect	Drive Shaft Seals	Inside Diameter: 1.500" Outside Diameter: 2.561" Seal Type: TB Material: NITRILE Thickness: 0.313"	36678506	1	\$6.31	\$6.31
3	MSCdirect	Fastener O-rings	Buna-N AS568A: Inside Diameter: 0.188" Thickness: 3/32 Outside Diameter: 0.375"	75748293	1	\$2.16	\$2.16
4	McMaster Carr	Radial Bearings	Steel Needle-Roller Bearing Roller for 1/2" Shaft Diameter, 1.1716" OD, 1/2" Width	5905K23	2	\$8.00	\$16.00
5	McMaster Carr	Thrust Bearings	Steel Tapered-Roller Bearing Roller Assembly for 1-1/8" Shaft Diameter	5709K22	2	\$23.17	\$46.34
6	McMaster Carr	Thrust Bearing Outer Races	Steel Tapered-Roller Bearing Outer Ring for 1-1/8" Shaft Dia	5709K59	2	\$7.22	\$14.44
7	FabroCast	Slip Rings	Slip Ring and Brush Assembly	1908RC-4BR-FAG150	1	\$800.00	\$800.00
8	McMaster Carr	Radial Bearing Housing Material	Housing for two radial bearings and one shaft seal.	8949K783	1	\$77.50	\$77.50
9	McMaster Carr	Thrust Bearing Housing Material	Housing for two thrust bearings and two shaft seals.	Same as Line 7	0	\$0.00	\$0.00
10	McMaster Carr	Gage Housing Material	Housing for strain gage assembly and two bearing assemblies.	89495K163	1	\$180.00	\$180.00
11	McMaster Carr	Drive Shaft Material	Shaft that connects the motor and strain gage assembly.	8934K231	1	\$40.00	\$40.00
12	McMaster Carr	Propeller Shaft Material	Shaft that connects the strain gage assembly and propeller/turbine	8984K673	1	\$40.00	\$40.00
13	MIT Central Machining	Custom Axle Nut	Nut to hold thrust bearing assy	NA	1		
14	MIT Central Machining	Custom Star Washer	Washer to lock axle nut	NA	1		
15	McMaster Carr	Set Screws	Drain plug set screw	99553A447	1	\$5.17	\$5.17
16	McMaster Carr	Heat Shrink Tubing	Tubing to cover sensitive electrical connections	8195K21	1	\$3.74	\$3.74
17	RaeTech	Amplifier Interface	Programming interface to set amplifier characteristics	1169-01-50-300	1	\$445.00	\$445.00
18	RaeTech	Amplifiers	Strain gage signal amplifiers	1169-01-50-100-H	2	\$395.00	\$790.00
19	Axis New England	Motor	Brushless DC Frameless Servo Motor	K095300-712	1	\$927.00	\$927.00
20	Axis New England	Controller	Coppy Xenus all Digital Servo Drive (240V/AC-40A)	XTL-230-40	1	\$845.00	\$845.00
21	Axis New England	Connector Kit	Servo Drive Solder Cup Connector Kit	XTL-CK	1	\$58.00	\$58.00
22	Axis New England	Cable Kit	Serial Cable Kit for PC Connection	SER-CK	1	\$20.00	\$20.00
23	Axis New England	Heat Sink	Standard Profile Heat Sink Kit	XTL-HS	1	\$68.00	\$68.00
24	Axis New England	Edge Filter	Xenus Edge Filter Accessory	XTL-FA-01	1	\$418.00	\$418.00
25	Axis New England	Edge Filter Connectors	Xenus Edge Filter Connector Kit	XTL-FK	1	\$12.00	\$12.00
26	McMaster Carr	Threaded Rod	4-40 Threaded Studs for Sensor/Shaft Attachment	95412A668	1	\$4.69	\$4.69
27	McMaster Carr	Nuts	4-40 Nuts for Sensor/Shaft Attachment	91841A005	1	\$2.88	\$2.88
28	McMaster Carr	Sealing Fastener	Socket Head Shoulder Cap Screw	93985A534	9	\$2.47	\$22.23
29	McMaster Carr	Drain Fastener	Socket Head Shoulder Cap Screw	95198A410	2	\$2.13	\$4.26
30	McMaster Carr	O-Rings	Sleeve o-rings (2mm x 8mm OD)	9262K715	1	\$4.41	\$4.41
31	McMaster Carr	O-Rings	Radial and Thrust Bearing Housing o-rings (2.943OD)	50187217	1	\$9.30	\$9.30
32	McMaster Carr	Sleeve Material	Stainless for Sleeve 3.50D x 2.50ID	89495K451	1	\$159.83	\$159.83
33	McMaster Carr	Propeller Nut	1/2" Hex Nut for Propeller Shaft	91831A140	1	\$7.64	\$7.64
34	McMaster Carr	Slip Ring Shaft Matl	Stainless for Slip Ring Shaft (1.5" x 18")	89535K653	1	\$121.85	\$121.85
35	McMaster Carr	Sealing Fastener O-ring	1/4"ID o-rings for use on sealing fasteners	2418T116	1	\$10.00	\$10.00
36	McMaster Carr	End Bearing	Ball Bearing on Motor End	2342K189	1	\$0.00	\$0.00
37	McMaster Carr	Transition Matl	Stainless 4"OD x 12" Long Solid	89535K631	1	\$151.79	\$151.79
38	McMaster Carr	End Bearing Housing Matl	Stainless Plate (1" x 4")	Same as Line 7	1	\$0.00	\$0.00
39	McMaster Carr	Encoder Adapter Matl	Stainless Rod (2.5" x 3")	Same as Line 8	1	\$0.00	\$0.00
40	McMaster Carr	Encoder Reader Mount Plate	Stainless Plate (1/4" x 4")	Same as Line 9	1	\$0.00	\$0.00
41	McMaster Carr	Transition Fastener	Socket Head Cap Screw - Long	91251A906	3	\$6.25	\$18.75
42	McMaster Carr	Brush Mount Plate Fastener	Socket Head Shoulder Cap Screw	93996A531	2	\$3.65	\$7.30
43	McMaster Carr	Motor Housing Matl	Stainless 4"OD x 2.5"ID x 12" Long Pipe	89495K631	1	\$259.54	\$259.54
44	McMaster Carr	O-Rings	End Bearing Housing o-rings (1/16" x 1/16" 3.5"OD)	9262K715	2	\$0.00	\$0.00
45	McMaster Carr	O-Rings	Transition o-rings (1/16" x 1/16" 3.75"OD)	9452K133	1	\$7.23	\$7.23
46	McMaster Carr	Propeller Pin	3/16" Straight Pin for Propeller Shaft	90145A514	1	\$11.05	\$11.05
47	McMaster Carr	Dyno Pipe	Stainless Precision Pipe (1.5"OD x 1.25"ID x 4ft)	89495K684	1	\$147.45	\$147.45
48	Automation Direct	Zero Phase Reactor	Zero Phase Reactor (RF Filter)	RF220X00A	2	\$25.50	\$51.00
49	Automation Direct	Panel Disconnect	Panel Disconnect	SD1-025-RR	1	\$22.50	\$22.50
50	Automation Direct	Panel Disconnect Remote Handle, Red-Yellow	Panel Disconnect Remote Handle, Red-Yellow	SD-HRY	1	\$5.75	\$5.75
51	Automation Direct	Panel Disconnect Remote Shaft, 300 mm	Panel Disconnect Remote Shaft, 300 mm	SD-S300	1	\$4.75	\$4.75
52	Automation Direct	NEWA 12 Enclosure	NEWA 12 Enclosure	B161410CH	1	\$123.00	\$123.00
53	Automation Direct	Enclosure Panel	Enclosure Panel	P1614	1	\$10.75	\$10.75
54	Automation Direct	24 VDC Power Supply, 30 W	24 VDC Power Supply, 30 W	PSC-24-030	1	\$56.75	\$56.75
55	Automation Direct	Supplementary Protector 1 Pole 6A C Curve	Supplementary Protector 1 Pole 6A C Curve	WMSTO06	1	\$9.25	\$9.25
56	Automation Direct	Fuse Holder, 1/16" I/PKG, M Class, 2 Pole, DIN MT, 30A	Fuse Holder, 1/16" I/PKG, M Class, 2 Pole, DIN MT, 30A	CHM2D-1	2	\$16.25	\$32.50
57	Automation Direct	Supplementary Protector 2 Pole 20A D Curve	Supplementary Protector 2 Pole 20A D Curve	WMS2D20	1	\$18.75	\$18.75
58	Automation Direct	Braking Resistor, 1000 W, 50 Ohm, NEMA 1 End.	Braking Resistor, 1000 W, 50 Ohm, NEMA 1 End.	GS-4015-BR-ENC	1	\$219.00	\$219.00
59	FilterConcepts	EMI Line Filter	EMI Line Filter	SF70L	1	\$66.13	\$66.13
60	McMaster Carr	Long Fastener	Transition Motor Housing Fastener	92196A287	3	\$3.68	\$11.04
61	McMaster Carr	Customized - Turned Head	Brush Plate Fastener	93996A531	2	\$3.65	\$7.30
62	McMaster Carr	Insert for nose fairing	Threaded Insert	92066A036	1	\$4.92	\$4.92

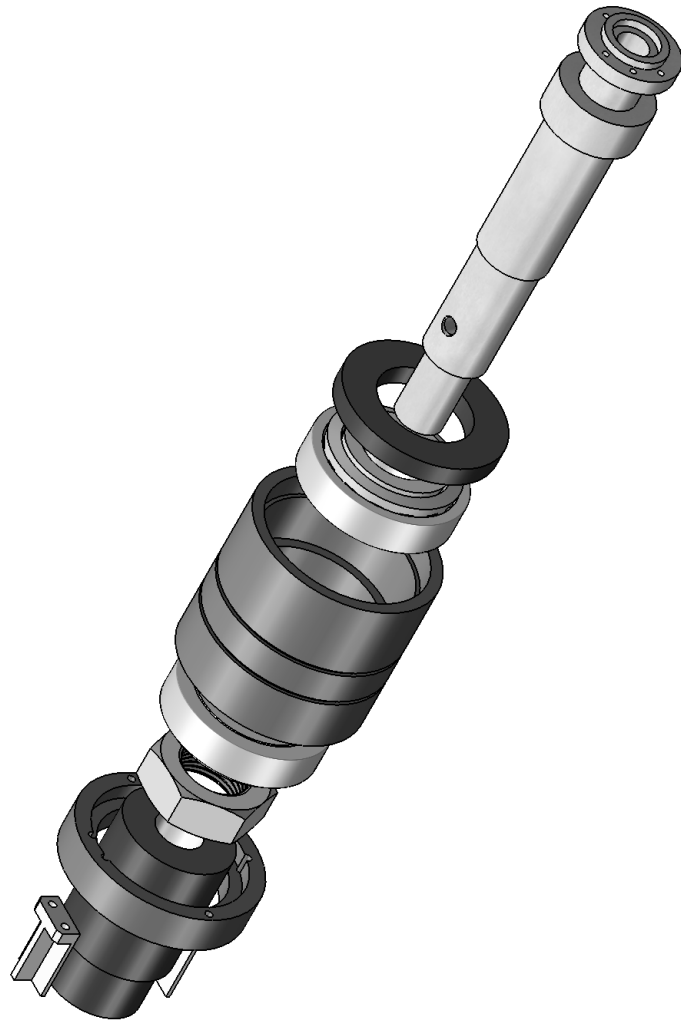
63	McMaster Carr	Customized - Shortened Threaded Stud for Nose Fairing	Motor Set Screws				
64	McMaster Carr	Threaded Stud for Nose Fairing	Fairing Fastener	92313A532	6	\$3.77	\$3.77
65	McMaster Carr	Snap Ring	Snap Ring for Motor Rotor	95412A847	1	\$3.10	\$3.10
66	McMaster Carr	O-rings	Soft Buna-N O-Ring ASS68A Dash Number 152, packs of 10	98585A118	1	\$4.27	\$4.27
67	McMaster Carr	DIN Rail	DIN Rail Steel, DIN 3, 35 mm Width, 7.5 mm H, 2 Meter L	24181164	1	\$9.67	\$9.67
68	McMaster Carr	Rotor Pin	0.25x1.5 rotor anti rotation pin	9981K16	1	\$8.50	\$8.50
69	McMaster Carr	Holding Fastener	Type 416 SS Precision Hex Socket Shldr Screw 1/4" Shoulder Dia, 3/8" L Shoulder, 10-32 Thread	97395A494	1	\$10.17	\$10.17
70	McMaster Carr	Filter/Guard	EmiRfH Fan Guard for 4.69" (119 mm) Fan	93955A535	6	\$2.86	\$15.96
71	McMaster Carr	Fans	AC Equipment Cooling Fan 4.69" SQ X 1" Depth, 80 CFM, 230 VAC	19159K38	3	\$5.80	\$17.40
72	McMaster Carr	Customized - Shortened	Type 302 Stainless Steel Compression Spring 1" Length, 5/32" OD, .016" Wire Diameter, packs of 6	1976K96	3	\$25.41	\$76.23
73	McMaster Carr	Adhesive	Loctite® 609 Retaining Compound, .02 oz, Green	1986K1	1	\$7.87	\$7.87
74	McMaster Carr	Adhesive Activator	Loctite® Primer No. 7471 Primer T, 1.75-Ounce Bottle	91458A23	1	\$1.71	\$1.71
75	McMaster Carr	Fasteners	Type 416 SS Precision Hex Socket Shldr Screw 1/4" Shoulder Dia, 1/4" L Shoulder, 10-32 Thread	66205A39	1	\$17.24	\$17.24
76	McMaster Carr	Fuses	13/32" Diameter Fast-Acting Fuse 15 Amps, 600 VAC, Fuse Types: Ktk, Atm, Ktk	93985A534	5	\$2.47	\$12.35
77	McMaster Carr	Power Cable	Flexible Multiconductor Cable Shielded, 14/4 Avg, 47" OD, 600 VAC, Gray	74505K783	4	\$10.61	\$42.44
78	McMaster Carr	Supply Power Connectors	Five Blade NEMA Style Turn-Lock Device L23-30, Male Plug, 347/600 VAC, 30 Amp	9936K75	15	\$6.89	\$103.35
79	McMaster Carr	Supply Power Connectors	Five Blade NEMA Style Turn-Lock Device L23-30, Male Plug, 347/600 VAC, 30 Amp	7184K47	1	\$27.74	\$27.74
80	McMaster Carr	Motor Power Plug	Four Blade NEMA Style Turn-Lock Device L15-30, Female Connector, 347/600 VAC, 30 Amp	7184K48	1	\$57.42	\$57.42
81	McMaster Carr	Signal Cable	Heavy Duty Category 9E Cable, Shielded	7164K49	1	\$22.82	\$22.82
				9284111	18	\$0.87	\$15.66
				Total	154	\$6,136.08	\$6,885.30

Notes:

RaeTech will solder wires to amplifier if requested
Total cost does not reflect manufacture costs
Power cable length was insufficient - needed another 8 ft

Appendix C – Drawings





TITLE:

Thrust Bearing Assembly Exploded

SIZE	Drawn By:	REV
A	Jerod Ketcham <i>j.ketcham@mit.edu</i>	0

SCALE: 1:2	Matl: 303 S.S.	SHEET 1 OF 1
------------	----------------	--------------

1

2

3

4

5

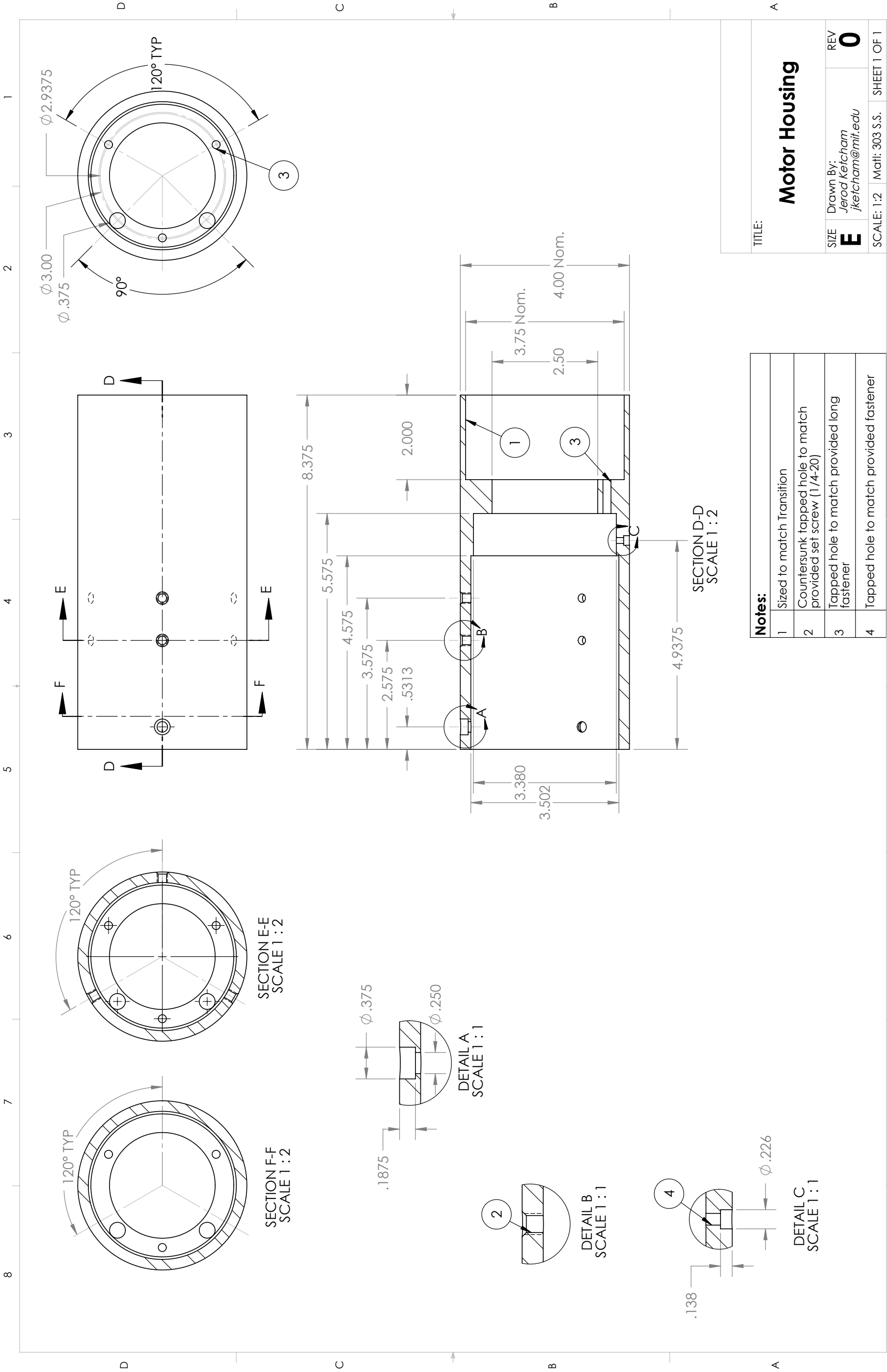


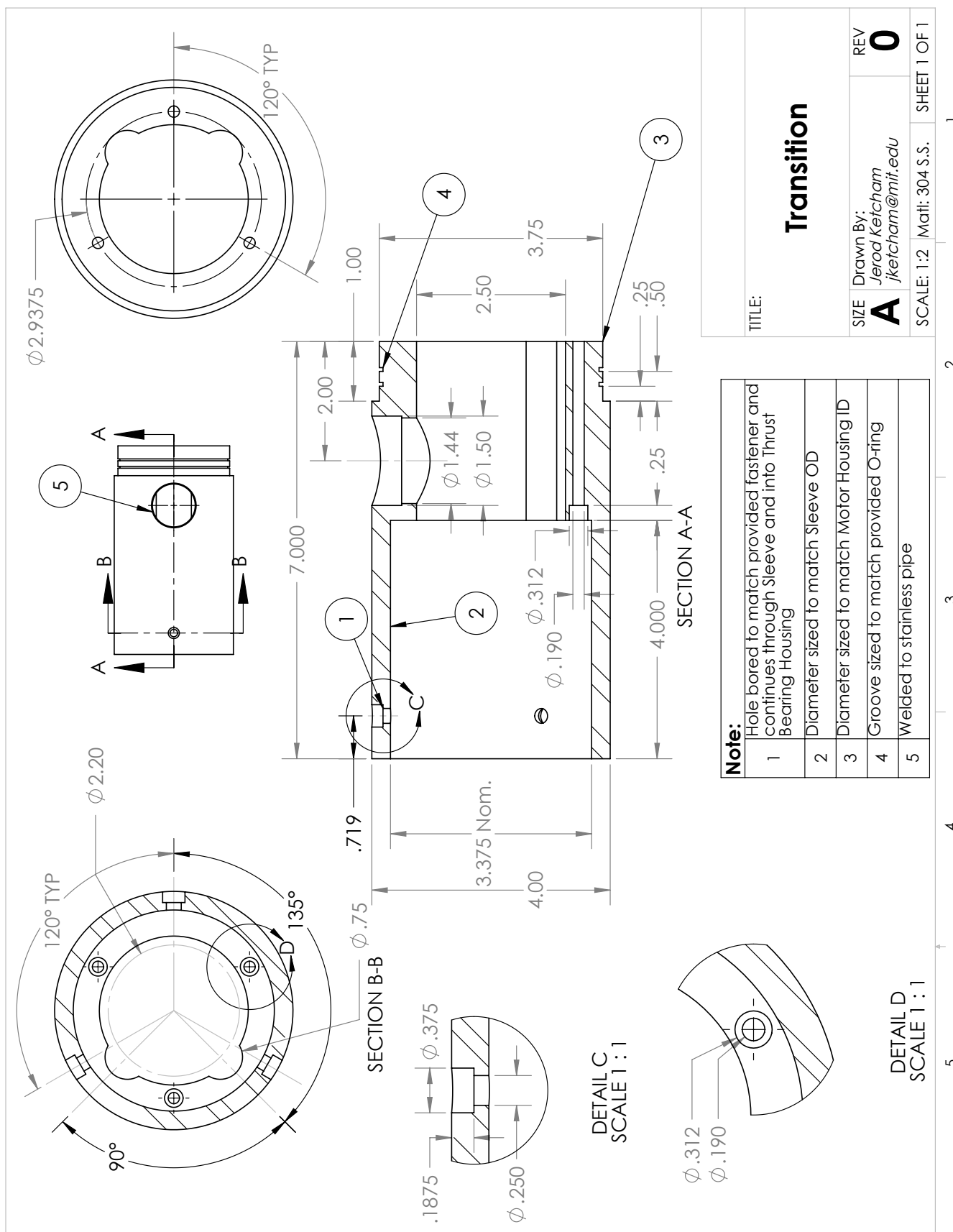
TITLE:

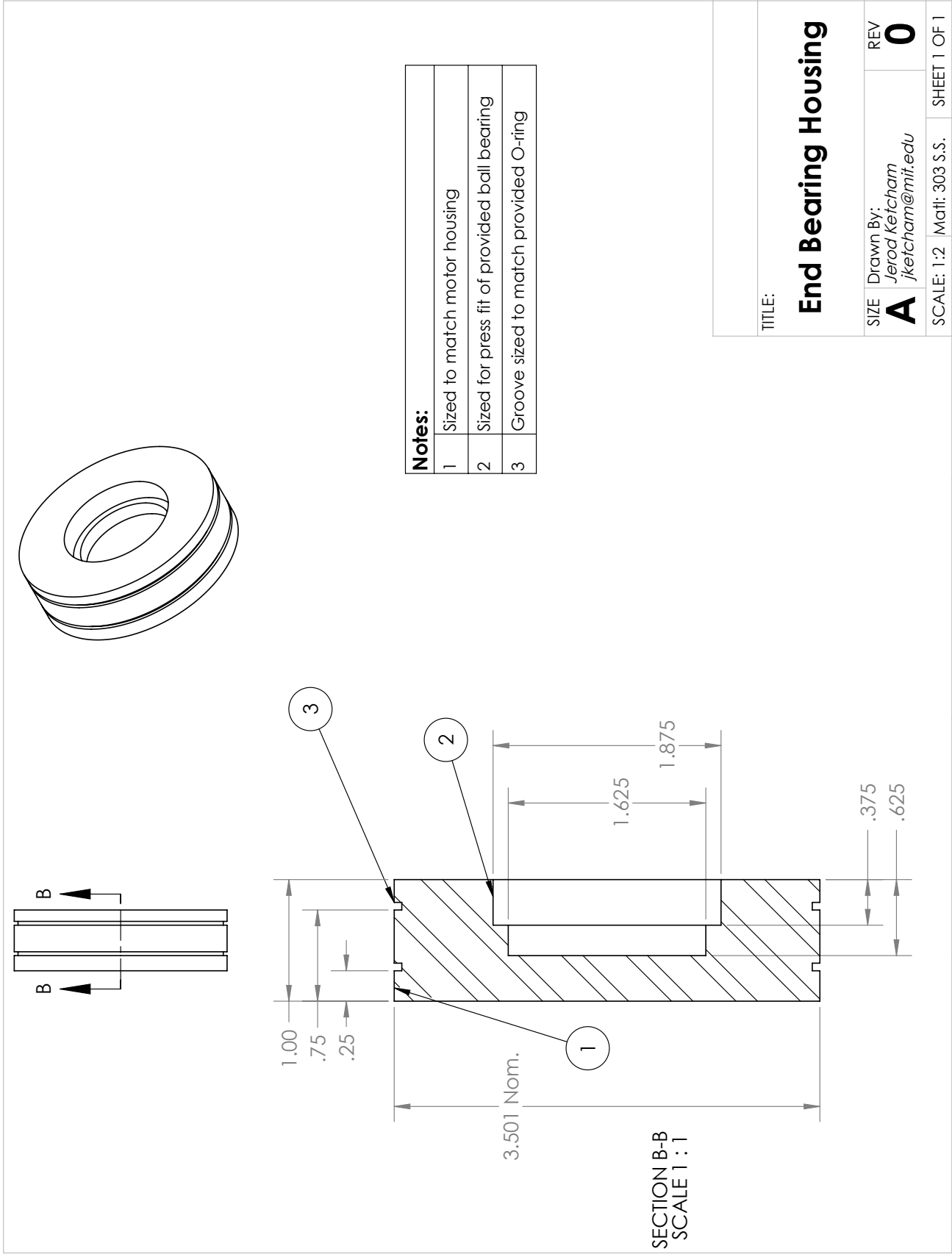
Radial Bearing Assembly Exploded

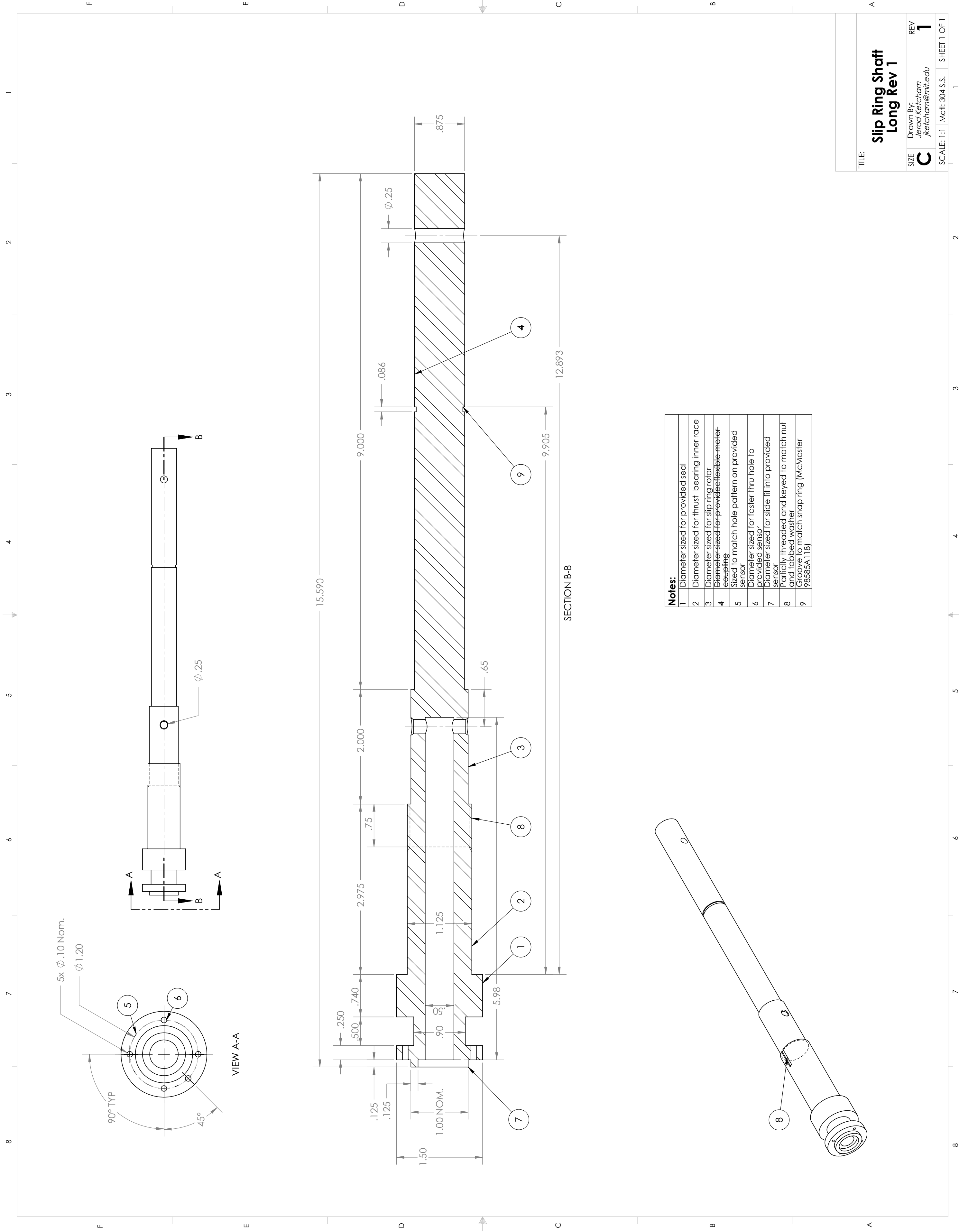
SIZE	Drawn By:	REV
A	<i>Jerod Ketcham</i> <i>jkeitcham@mit.edu</i>	0

SCALE: 1:2 | Matl: 303 S.S. | SHEET 1 OF 1

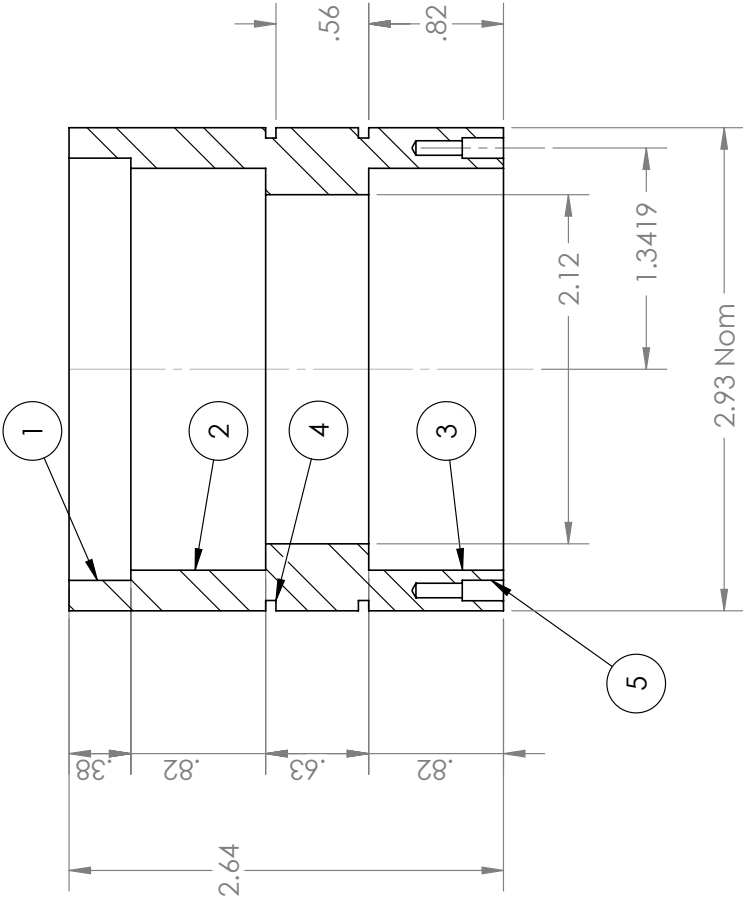




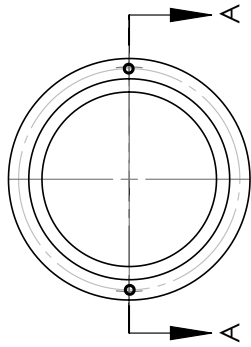
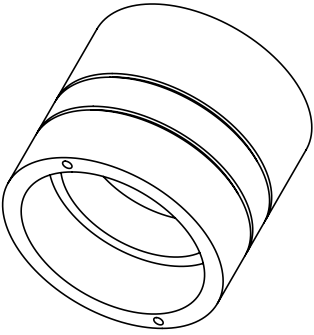




SECTION A-A
SCALE 1 : 1



Notes:	
1	Diameter sized for provided seal
2	Diameter sized for provided bearing
3	Diameter sized for provided bearing
4	Groove to match provided oring
5	Hole to match provided fastener (McMaster 93996A531)



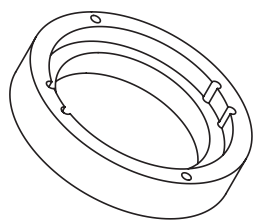
TITLE:

Thrust Bearing Housing

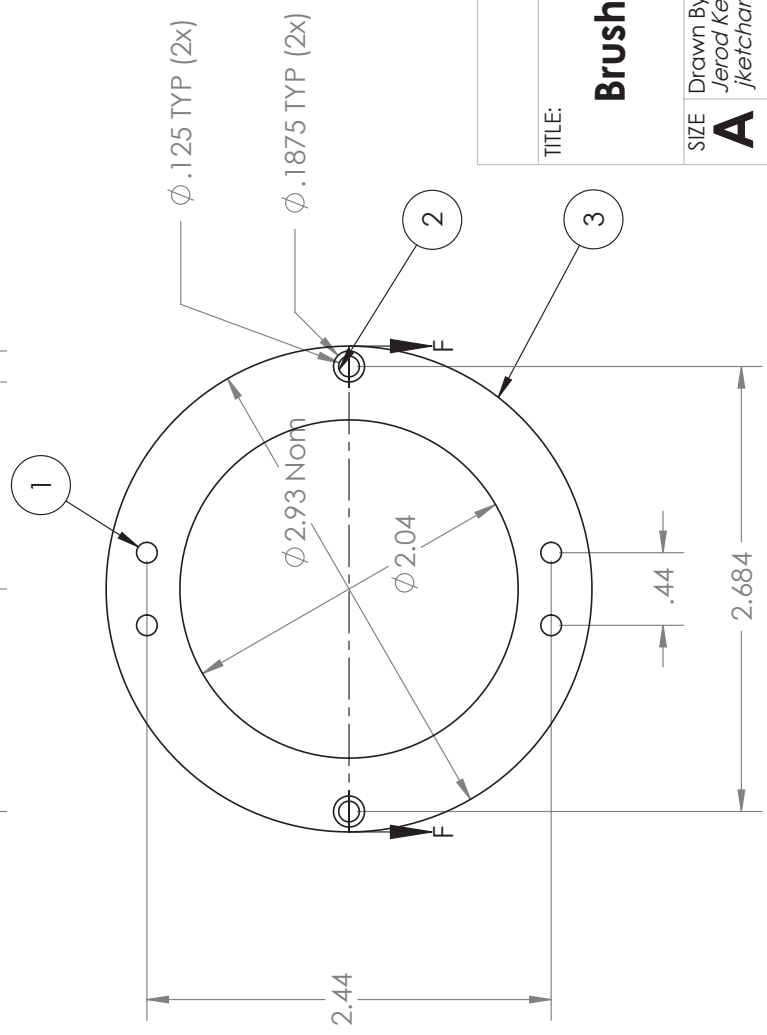
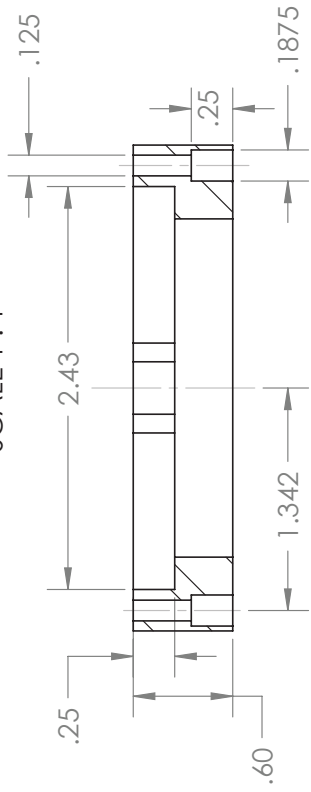
SIZE	Drawn By:	REV
A	<i>Jerod Ketcham</i> <i>jketcham@mit.edu</i>	0

SCALE: 1:2 Matl: 303 S.S. SHEET 1 OF 1

Notes:	
1	Tapped hole to match provided brush block
2	Thru hole to align with hole in Thrust Bearing Housing
3	OD to match Thrust Bearing Housing



SECTION F-F
SCALE 1 : 1

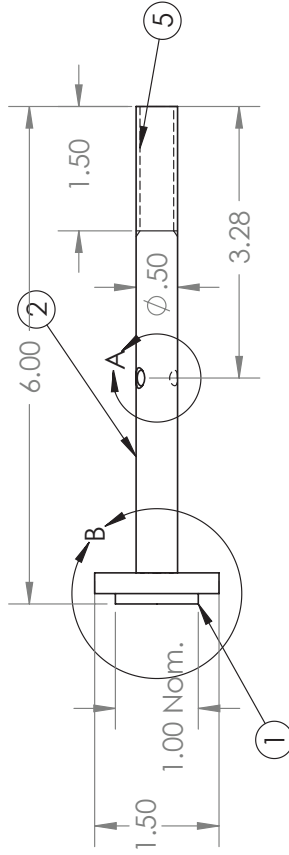
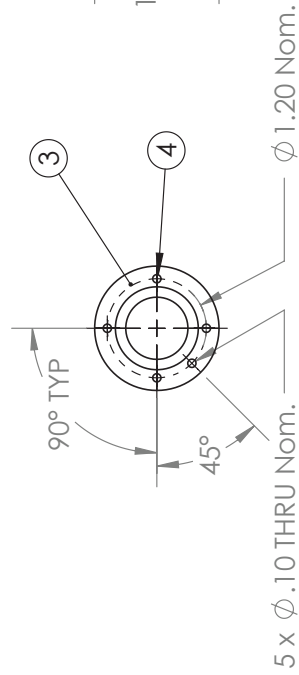
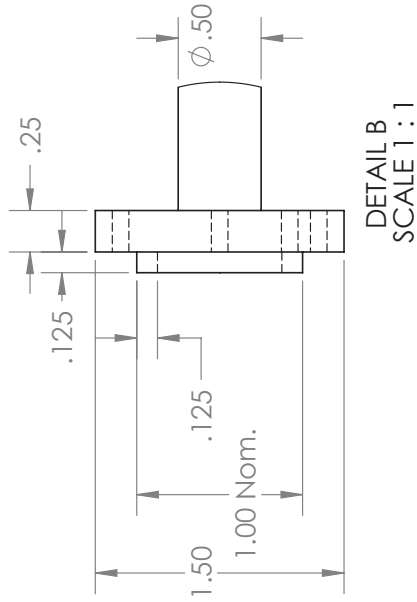
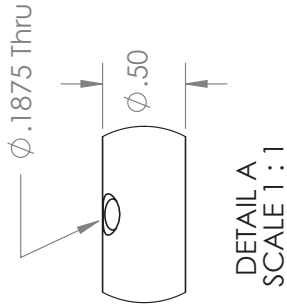
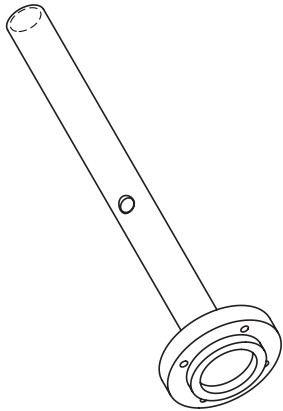


TITLE:

Brush Mount Plate

SIZE	Drawn By:	REV
A	Jerod Keitcham jkeitcham@mit.edu	0

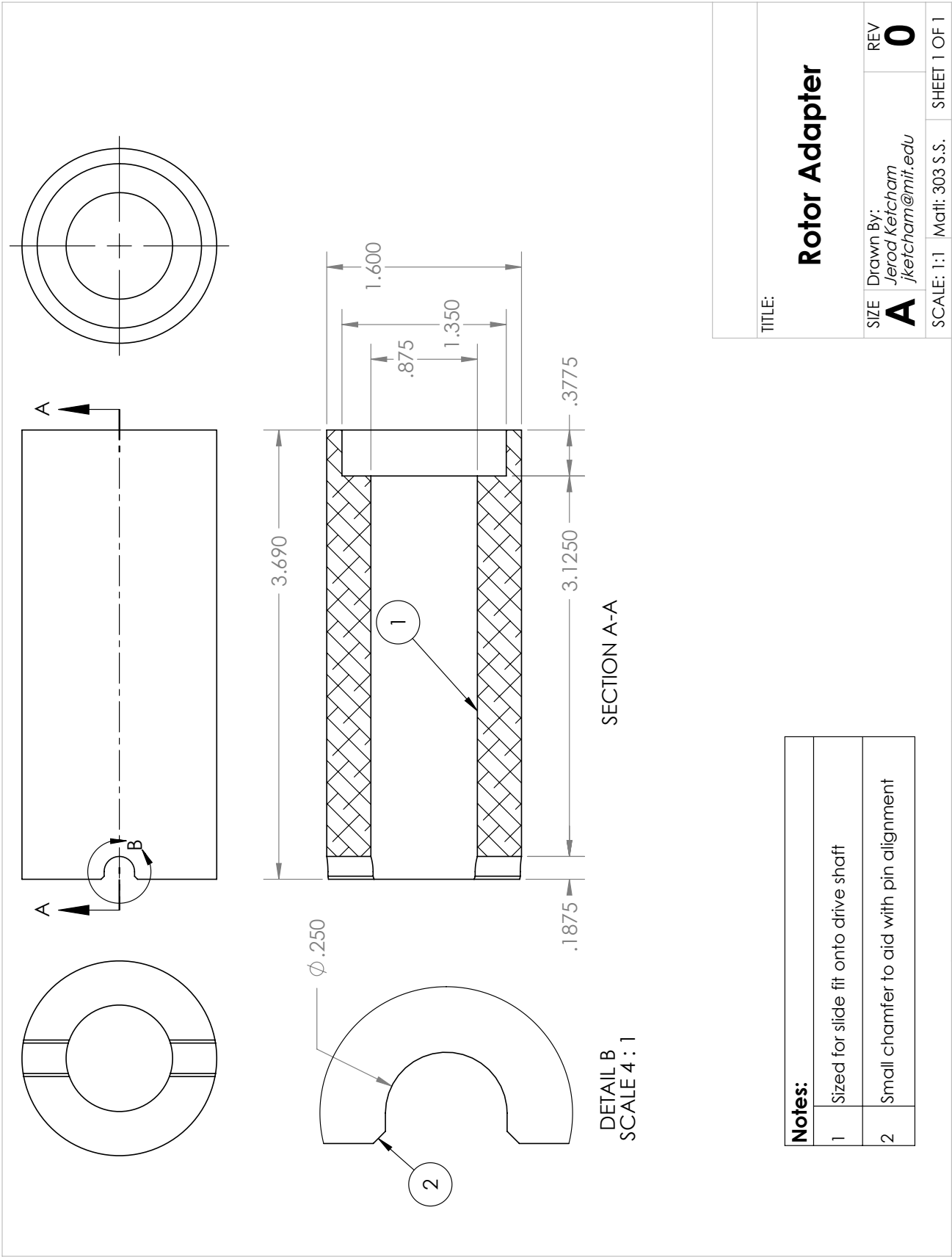
SCALE: 1:1	Matl: 303 S.S.	SHEET 1 OF 1
------------	----------------	--------------

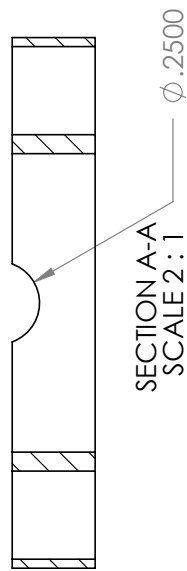
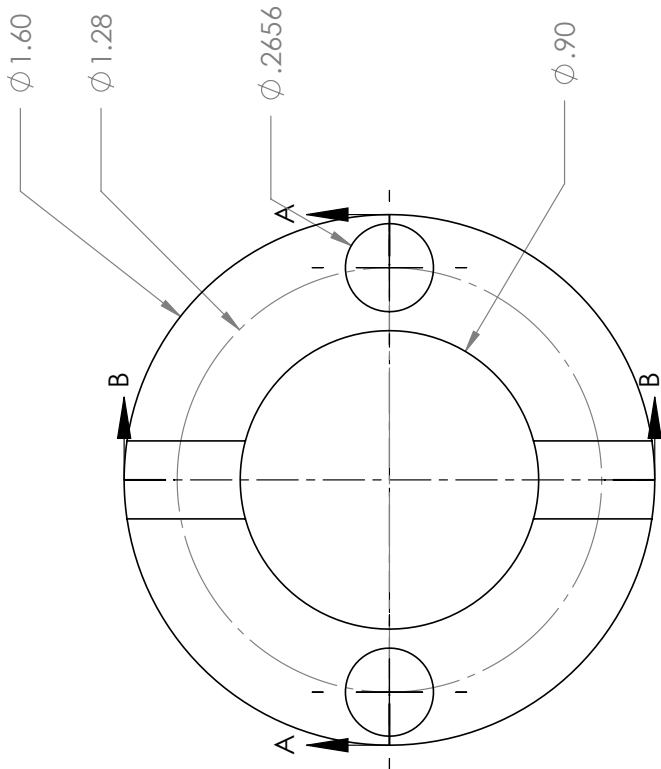
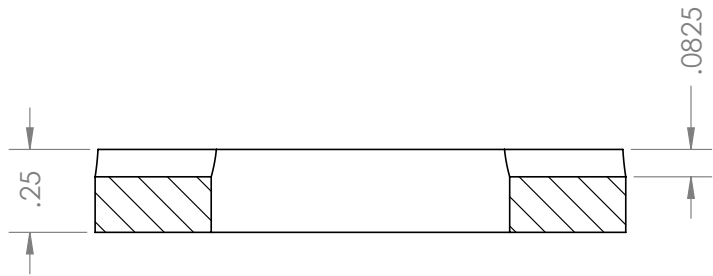


Notes:	
1	Sized for sliding fit into provided sensor
2	Sized for sliding fit into provided needle bearing
3	Sized to match hole pattern on provided sensor
4	Diameter sized for thru hole for 4-40 fastener
5	Threaded to match provided nut 1/2-20

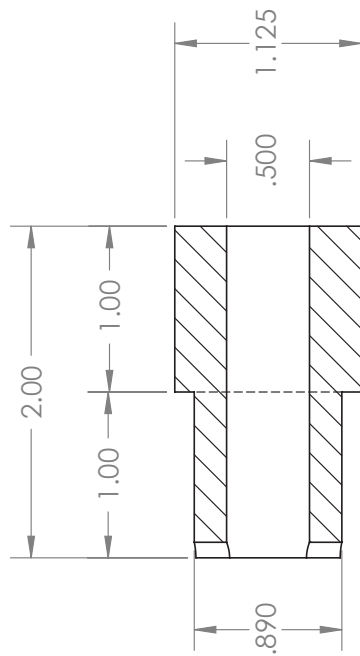
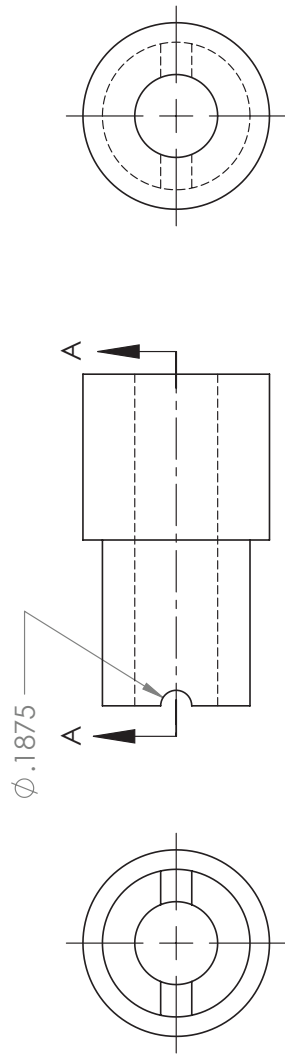
TITLE:		
Sensor Shaft		
SIZE	Drawn By: Jerod Ketcham jketcham@mit.edu	REV 0
SCALE: 1:2		Matl: 303 S.S.
		SHEET 1 OF 1

1 2 3 4 5





TITLE:			
Rotor Plate			
SIZE	Drawn By:	REV	
A	Jerod Ketcham jketcham@mit.edu	0	
SCALE: 1:1		Matl: 303 S.S.	SHEET 1 OF 1



SECTION A-A

TITLE:

Hub Adapter

SIZE	Drawn By:	REV
A	Jerod Ketcham jketcham@mif.edu	0

SCALE: 1:1 Matl: 6061-T61 SHEET 1 OF 1

